THREE-DIMENSIONAL ELASTO-STATIC PROBLEMS USING ISOPARAMETRIC FINITE ELEMENTS

Dale Eugene Hanson

United States Naval Postgraduate School

THESIS

THREE-DIMENSIONAL ELASTO-STATIC PROBLEMS
USING
ISOPARAMETRIC FINITE ELEMENTS

by

Dale Eugene Hanson

Thesis Advisor: G. Cantin

June 1971

Three-Dimensional Elasto-Static Problems Using Isoparametric Finite Elements

by

Dale Eugene Hanson
Lieutenant, United States Navy
B.S., United States Naval Academy, 1964

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1971

ABSTRACT

The objective of the project described in this report was to develop a computer system which would generate the required input data for a structural analysis of three-dimensional elasto-static problems using isoparametric finite elements. Element connectivity, nodal point coordinates, consistent gravity loads, and consistent pressure loads are generated. A variety of algorithms are used in the system to reduce the amount of input data required. The computer system and a sample problem are discussed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

The author wishes to express his appreciation to Professor Gilles Cantin who introduced this author to the finite element method of structural analysis and to the concept of an automatic mesh generator and who provided his moral support, encouragement, and assistance throughout the development of this project.

I. INTRODUCTION

For a structural analysis of any non-trivial elasto-static problem by the finite element method, a large amount of data must be provided to the computer system which performs the calculations. The required information includes the arrangement of nodes within each element, the coordinates of all nodes, and the applied loads. After the numerical values of this data have been determined, which is a large problem in itself, the information must be supplied to the computer system. This is normally accomplished by punching the appropriate data on computer cards. The typing of these cards is also a difficult and time-consuming job. More important, however, is the ever-present possibility of human errors during the performance of these tasks. If an error is not detected prior to submitting the data for processing, expensive computer time would be wasted.

With the above considerations in mind, it is readily apparent that a computer system, which would generate most of the data required by the primary computer system for a complete structural analysis, would be very useful. In addition, this mesh generator should require as little input data as possible.

References 5, 8, and 10 contain reports of efforts which have been directed at coding computer systems which will aid the analyst in the interpretation of output data. A few computer systems, which are reported in Refs. 1, 2, and 9, have been coded to automatically generate input data. However, these input data generation systems

are either for types of finite elements other than isoparametric elements or for two-dimensional elements.

Two input data generator systems for three-dimensional isoparametric finite elements will be presented in this thesis. One system is for quadratic elements and the second system is for cubic elements.

II. ISOPARAMETRIC FINITE ELEMENTS

The isoparametric concept will be discussed in this chapter in sufficient detail to establish only the procedures for constructing isoparametric elements and the validity of relationships used to obtain the consistent gravity and pressure loads. The readers are referred to Refs. 4 and 7 for further details of the isoparametric concept.

A. DISPLACEMENT FUNCTIONS

The displacements of nodes in any one element are obtained by using shape functions which are defined in a system of dimensionless coordinates (ξ,η,ζ) that are unique for that element. The coordinates range from minus one to plus one.

$$\begin{aligned} u(x, y, z) &= \sum_i N_i(\xi,\eta,\zeta)\, u_i \\ v(x, y, z) &= \sum_i N_i(\xi,\eta,\zeta)\, v_i \\ w(x, y, z) &= \sum_i N_i(\xi,\eta,\zeta)\, w_i \end{aligned} \tag{1}$$

where $N_i(\xi,\eta,\zeta)$ are the shape functions and u_i, v_i, w_i are the displacement components of node "i" in a global reference system (x, y, z), where "i" assumes values from 1 to the total number of nodes in the element. This global reference system is then related to the dimensionless coordinates by the same shape functions as follows:

$$\begin{aligned} x(\xi,\eta,\zeta) &= \sum_i N_i(\xi,\eta,\zeta)\, x_i \\ y(\xi,\eta,\zeta) &= \sum_i N_i(\xi,\eta,\zeta)\, y_i \\ z(\xi,\eta,\zeta) &= \sum_i N_i(\xi,\eta,\zeta)\, z_i \end{aligned} \tag{2}$$

where x_i, y_i, z_i are the coordinates of node "i" in the global reference system [4].

The shape functions referred to above have been obtained from [7]. They are listed in section 2.C.

B. TRANSFORMATIONS

In order to obtain the consistent gravity and pressure loads, the volumetric increment, dxdydz, must be found in terms of the dimensionless coordinates. That operation requires the Jacobian matrix [4].

$$[J] = \begin{bmatrix} \partial x/\partial \xi & \partial y/\partial \xi & \partial z/\partial \xi \\ \partial x/\partial \eta & \partial y/\partial \eta & \partial z/\partial \eta \\ \partial x/\partial \zeta & \partial y/\partial \zeta & \partial z/\partial \zeta \end{bmatrix} \qquad (3)$$

The element of volume is then transformed as follows:

$$dV = \det[J] d\xi \; d\eta \; d\zeta \qquad (4)$$

C. SHAPE FUNCTIONS

In this listing of the shape functions, attention is directed to the tri-quadratic and tri-cubic elements, for which the mesh generator programs were written.

1. Quadratic Elements

See Figure 1 for identification of the nodes.

Corner nodes: 1, 3, 5, 7, 13, 15, 17, and 19.

$$N_i = (1/8)(1 + \xi_o)(1 + \eta_o)(1 + \zeta_o)(\xi_o + \eta_o + \zeta_o - 2) \qquad (5)$$

where $\xi_o = \xi_i \xi$ and ξ_i is ± 1; similarly for η_o and ζ_o.

Mid-side nodes: 2, 6, 14, and 18.

$$N_i = (1/4)(1 - \xi^2)(1 + \eta_o)(1 + \zeta_o) \qquad (6)$$

Mid-side nodes: 4, 8, 16, and 20.

$$N_i = (1/4)(1 - \eta^2)(1 + \xi_o)(1 + \zeta_o) \tag{7}$$

Mid-side nodes: 9, 10, 11, and 12.

$$N_i = (1/4)(1 - \zeta^2)(1 + \xi_o)(1 + \eta_o) \tag{8}$$

2. <u>Cubic Elements</u>

See Figure 2 for identification of nodes.

Corner nodes: 1, 4, 7, 10, 21, 24, 27, and 30.

$$N_i = (1/64)(1 + \xi_o)(1 + \eta_o)(1 + \zeta_o)[9(\xi^2 + \eta^2 + \zeta^2) - 19] \tag{9}$$

Mid-side nodes: 2, 3, 8, 9, 22, 23, 28, and 29.

$$N_i = (9/64)(1 - \xi^2)(1 + 9\xi_o)(1 + \eta_o)(1 + \zeta_o) \tag{10}$$

where $\xi_i = \pm 1/3$, $\eta_i = \pm 1$, and $\zeta_i = \pm 1$.

Mid-side nodes: 5, 6, 11, 12, 25, 26, 31, and 32.

$$N_i = (9/64)(1 - \eta^2)(1 + 9\eta_o)(1 + \xi_o)(1 + \zeta_o) \tag{11}$$

where $\eta_i = \pm 1/3$, $\xi_i = \pm 1$, $\zeta_i = \pm 1$.

Mid-side nodes: 13, 14, 15, 16, 17, 18, 19, and 20.

$$N_i = (9/64)(1 - \zeta^2)(1 + 9\zeta_o)(1 + \xi_o)(1 + \eta_o) \tag{12}$$

where $\zeta_i = \pm 1/3$, $\xi_i = \pm 1$, and $\eta_i = \pm 1$.

Note that the convention for numbering of nodes within any element is to begin at the location where $\xi = \eta = \zeta = 1$ and to consecutively number the nodes, proceeding in a counter-clockwise direction around the ζ-axis (see Figures 1 and 2).

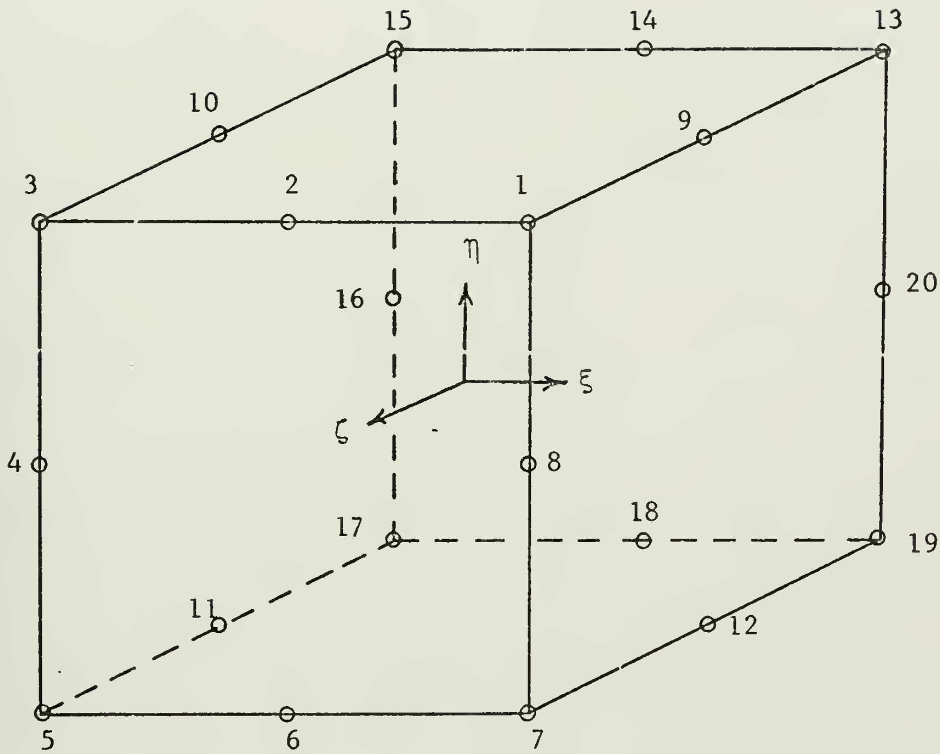


Figure 1. Quadratic Element

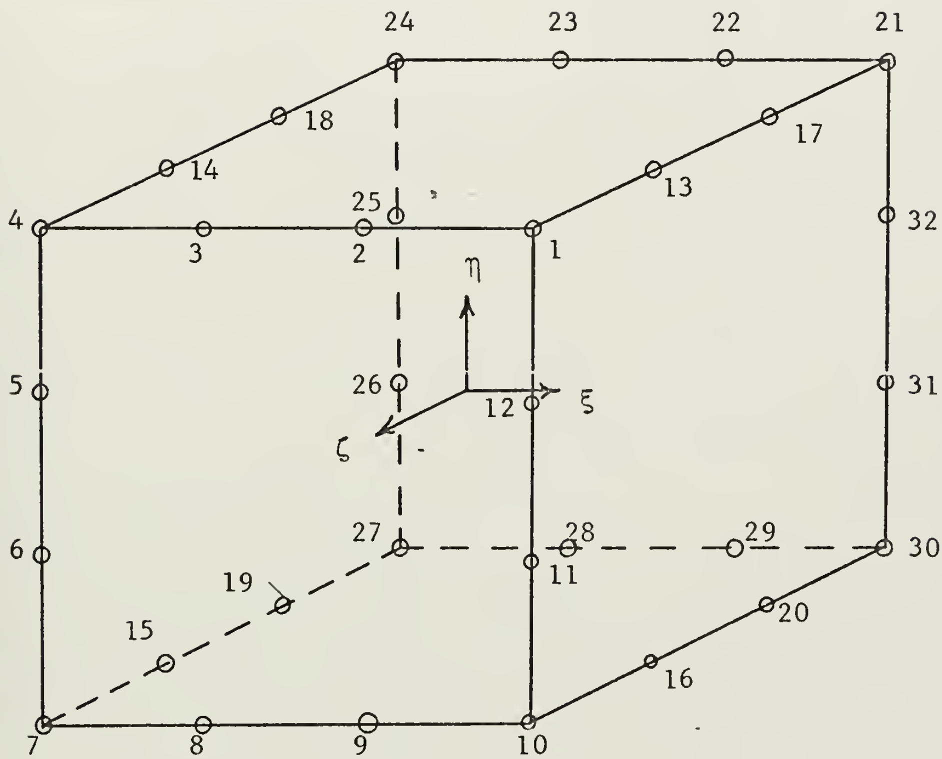


Figure 2. Cubic Element

III. DISCUSSION OF COMPUTER PROGRAMS

The computer programs presented in this thesis were written to support a computer system called TRISOP. TRISOP was coded by Professor G. Cantin of the U. S. Naval Postgraduate School, Monterey, California. TRISOP performs a structural analysis of three-dimensional elasto-static problems using isoparametric finite elements. Information required by TRISOP for any problem includes general mesh parameters, title cards, format statements, and material properties; Young's modulus of elasticity, Poisson's ratio, and the coefficient of thermal expansion. Fifteen cards are needed to supply this data to TRISOP. The bulk of the data required by TRISOP is the element connectivity, structure geometry, loading, and boundary conditions. The boundary conditions will vary with each problem; therefore, no effort has been made to automate this process.

The mesh generating computer programs discussed in this chapter compute the remainder of the data required by TRISOP: element connectivity, coordinates of nodal points, consistent gravity load vector, and consistent pressure load matrix. Additional information which is helpful in preparing an input deck of cards for TRISOP is also computed. One of these programs generates data for quadratic elements and the second program generates data for cubic elements. These programs will be called QUAMEG and CUMEG.

A. ELEMENT CONNECTIVITY

A structure of any shape can be visualized as a cubical piece of plastic material which can be stretched, compressed, and otherwise

appropriately deformed in such a way that it will assume the geometry of the structure. The element connectivity is determined when the structure is in this initial abstract cubical shape. The dimensionless coordinates of this abstract structure are ξ', η', ζ'.

Numbering of nodes is begun in a face which will have the fewest nodes. In that face, numbering proceeds sequentially along the edge having the smallest number of nodes as shown in Figure 3. This convention reduces the size of the half-bandwidth of the system, which directly affects the space and time required by the computer to solve the problem. The coordinate system is positioned such that node one is at $\xi' = 0$, $\eta' = 1$, and $\zeta' = 1$. An example is discussed in Appendix A.

For this mesh, the connectivities of the elements follow simple but tedious rules that were coded directly.

B. SOLUTION TIME AND SPACE REQUIREMENTS

The time required to solve the system of equations in the problem and the space required by the computer to store all data are important information. They must be determined before the problem is submitted for processing. The calculation of this information requires no additional input data and it is therefore included in QUAMEG and CUMEG to reduce the effort required of the programmer.

The time required to solve the equations is computed using an empirical formula which was constructed by Cantin [4]. This calculation has been included in the coding of QUAMEG and CUMEG.

The space required to store the various bits of information for a problem depends upon the total number of records and the

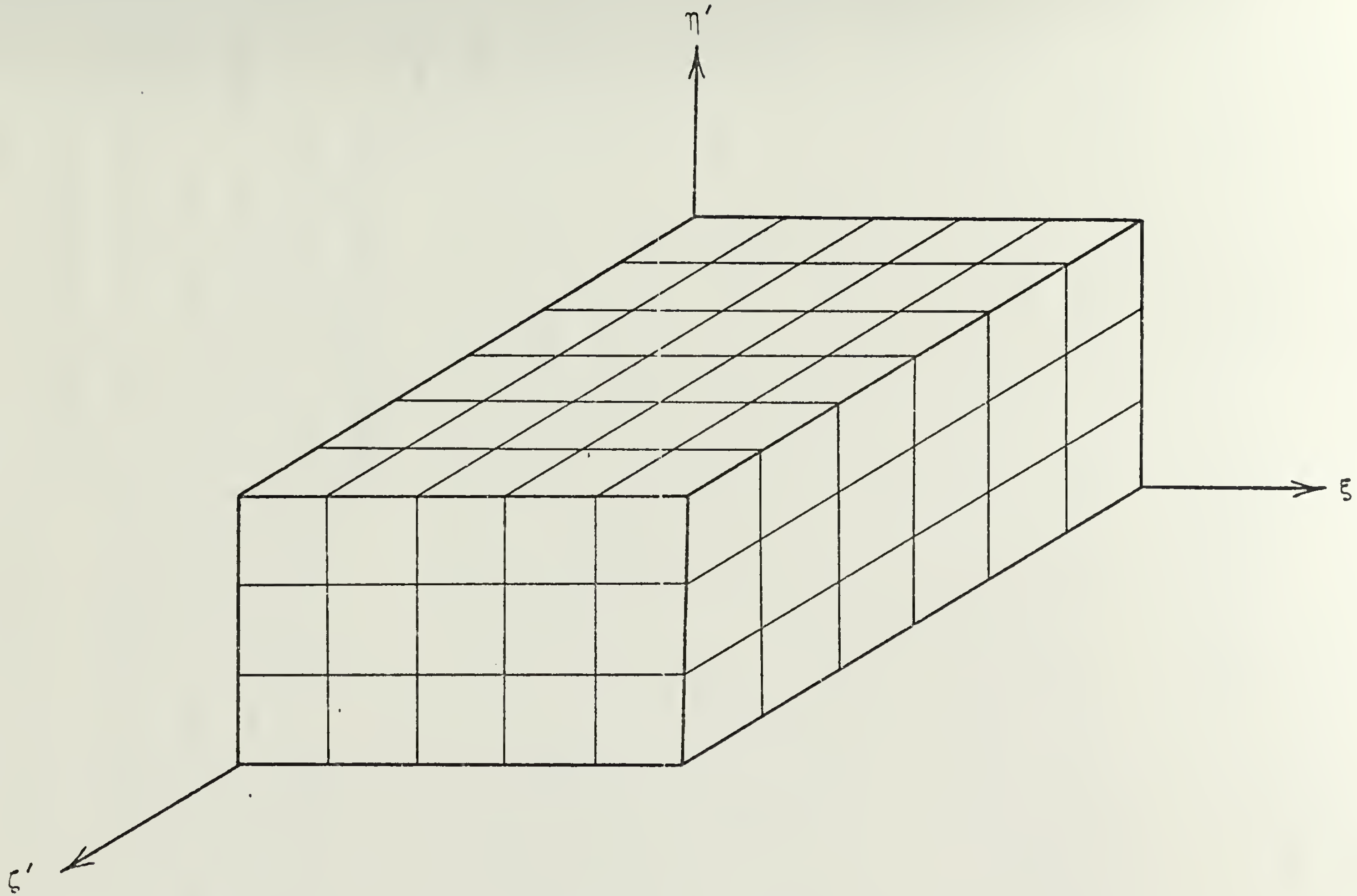


Figure 3. Abstract Structure

length of each record. These will vary with different types of computers. Simple formulas which generate the correct results for the IBM 360/67 computer used at the U. S. Naval Postgraduate School are employed in the coding of QUAMEG and CUMEG.

C. NODES ON BOUNDARY SEGMENTS

Before the coordinates of nodes which are within the boundaries of a structure can be computed, the coordinates of all corner nodes on the edges of the bounding surfaces must be specified. This problem is simplified by the method discussed below.

The term "boundary segment" must first be defined for the purpose of this discussion. A segment must satisfy a few simple conditions which depend upon the orientation of the proposed segment in the structure and upon the type of coordinate system used to describe the geometry of the structure. QUAMEG and CUMEG are coded to allow the use of either rectangular or cylindrical coordinate systems. A combination of the two coordinate systems in any one problem is not permitted.

1. Rectangular Coordinates

When rectangular coordinates are used, a segment is any straight line between two corner nodes of the mesh. The element divisions along this line must be uniform. There are no other restrictions imposed upon a segment when rectangular coordinates are used.

2. Cylindrical Coordinates

When cylindrical coordinates are used, the radius and reference angle are taken in planes parallel to the x-y plane. In this case a segment can be an arc of circle centered on z and in a plane

parallel to plane x-y, or a straight line in a plane containing the z-axis. Again, element divisions along these lines must be uniform.

In order to determine the coordinates of nodes along a boundary segment, the coordinates of the corner nodes at the ends of the segment must be provided in the input data. In addition, the location of the segment within the mesh must be specified. The method of specifying the location of a segment is described in sections 4.B.3 and 4.B.4.

In Figure 4, the vector AB is easily obtained from the coordinates of nodes A and B. After calculating the values of increments of distances along x, y, and z, the intermediate nodes are easily obtained.

If the bounding surfaces of a structure are of an irregular configuration such that portions of the edges of that surface cannot be specified by segments, the coordinates of all nodes along this irregular portion of the bounding surface must be provided in the input data.

D. CORNER NODES

Once the coordinates of the nodes on all of the boundary segments are established, the coordinates of corner nodes on the bounding surfaces and on interior slices are calculated using a method described below. The bounding surfaces are considered in the following order (Figure 3): face 1 ($\zeta'=1$), face 2 ($\zeta'=0$), face 3 ($\xi'=0$), face 4 ($\xi'=1$), face 5 ($\eta'=1$), face 6 ($\eta'=0$). Slices which are each of constant ζ' are then considered until the coordinates of all corner nodes are calculated.

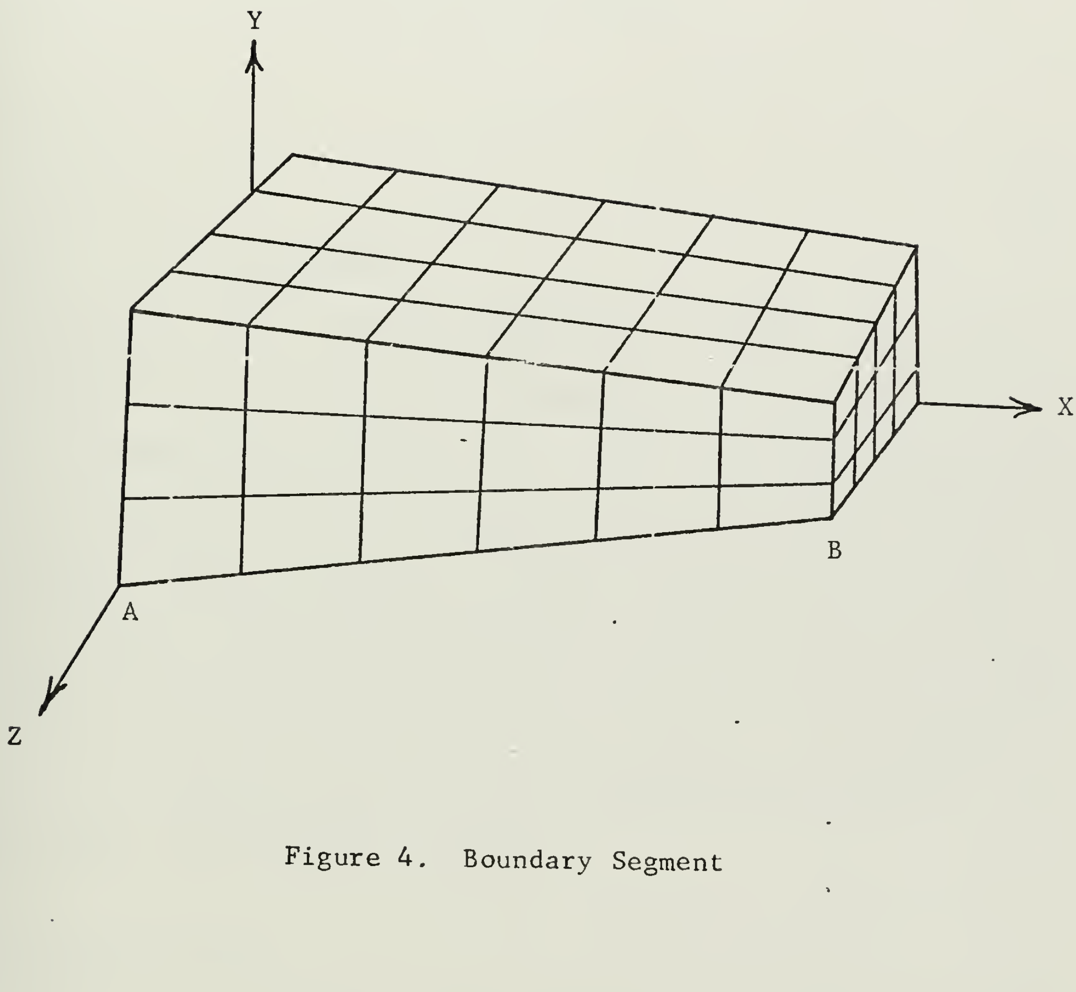


Figure 4. Boundary Segment

The method used for calculating the coordinates of corner nodes within the edges of a bounding surface or slice can best be understood with the example illustrated in Figure 5. In this example, the surface ABCD is the real surface for which a mesh is desired. The mesh is first constructed in the $\alpha\beta$ plane as shown.

At this time, the coordinates of all the corner nodes on the boundary are known. If the coordinates of the interior nodes are required to satisfy the Laplacian equation:

$$\nabla^2(x) = 0$$

$$\nabla^2(y) = 0$$

$$\nabla^2(z) = 0 \tag{13}$$

where $\nabla^2 = \partial^2/\partial\alpha^2 + \partial^2/\partial\beta^2$

a set of acceptable interior nodes will be obtained. The dummy variables (α,β) are replaced by appropriate combinations of ξ', η', and ζ', depending upon the location of the face being considered. For this computation, the standard Laplacian molecule of finite difference theory is used and the final coordinates are obtained by relaxation. Since the coordinates of the nodes within the edges of a face are arbitrary, the relaxation process can be stopped at any time, depending upon the degree of approximation desired. In this code, sixty iterations are used.

E. MID-SIDE NODES

After the coordinates of all corner nodes have been established, the coordinates of the mid-side nodes are computed by averaging the coordinates of adjacent corner nodes. Although the mid-side nodes need not be located exactly mid-way between the corner nodes, it is the simplest means of determining their coordinates.

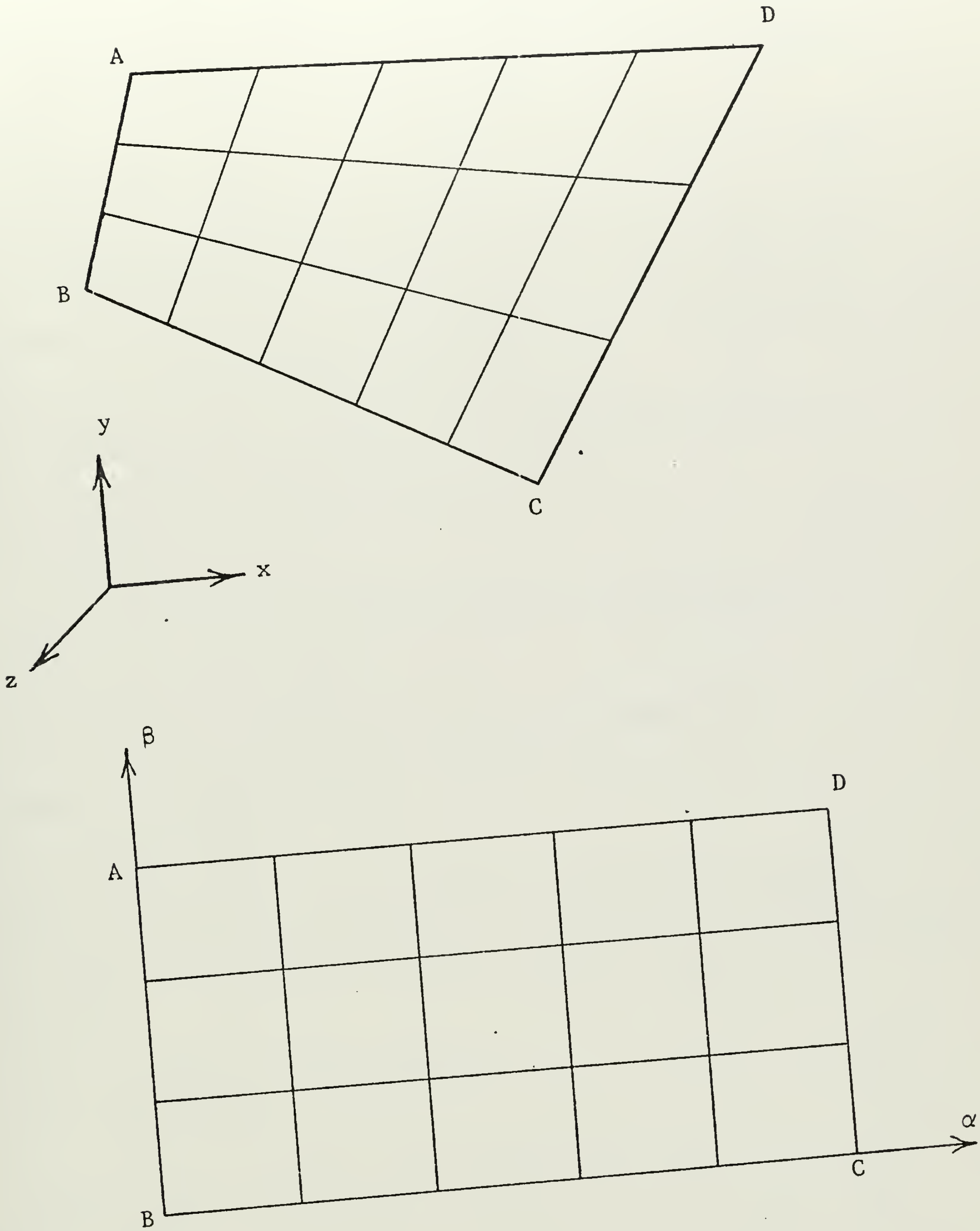


Figure 5. Bounding Surface

F. CONSISTENT GRAVITY LOAD VECTOR

The z-axis has been chosen as the line of action of the gravity load. The work done by the force due to gravitational acceleration on a body is

$$U = \int_V \rho g w dV \tag{14}$$

where U is the work done; ρ is the mass density of the body; g is gravitational acceleration; w is the displacement of the body in the z-direction; and dV is the element of volume.

The work done by concentrated forces acting at each node in the structure mesh is, in matrix notation:

$$U = \langle V \rangle \left\{ w_i \right\} \tag{15}$$

where $\langle V \rangle$ is the consistent gravity load vector and $\left\{ w_i \right\}$ are the respective node displacements.

From Chapter 2, recall the following two expressions:

$$w(x,y,z) = \sum_i N_i(\xi,\eta,\zeta)\left\{ w_i \right\}$$

$$dV = \det[J]\, d\xi d\eta d\zeta.$$

When these expressions are substituted into equation 14 and the result identified to equation 15, the consistent gravity load vector is obtained.

$$\langle V \rangle = \rho g \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \det[J]\langle N_i \rangle d\xi d\eta d\zeta \tag{16}$$

In this form, equation 16 is ideally suited for solution by Guassian cubature. In QUAMEG, four Gauss points are used in each of the three directions, for a total of 64 points within each element. Five Gauss points are used in CUMEG. The weight factors and coordinate values are obtained from Ref. 11. The Jacobian matrix is determined by using three function statements and the shape functions

are determined with two function statements in Subroutine GRAP (Appendix C). These function statements are permuted to obtain the partial derivatives of the shape functions and the values of the shape functions for all nodes in the mesh.

G. CONSISTENT PRESSURE LOAD MATRIX

The consistent pressure load matrix for a uniformly distributed pressure acting on any face of an element is determined in a manner similar to that used for the gravity load.

The work done by the uniformly distributed pressure is:

$$U = \int_A (-p\bar{n}\ dA)\cdot(\bar{u}) \tag{17}$$

where $\bar{n}$ is a unit normal vector (positive outward) and $\bar{u}$ is the displacement vector at the surface (see Figure 6). The displacement vector $\bar{u}$ can also be written as:

$$\bar{u} = u(x,y,z)\bar{i} + v(x,y,z)\bar{j} + w(x,y,z)\bar{k} \tag{18}$$

The vector $\bar{n}dA$ is found from differential geometry:

$$\bar{n}dA = (\partial\bar{r}/\partial\alpha)X(\partial\bar{r}/\partial\beta)\ d\alpha d\beta \tag{19}$$

where $\bar{r} = x\bar{i} + y\bar{j} + z\bar{k}$ is a position vector of a point on the pressurized surface and α and β are dummy variables, standing for either ξ, η, or ζ.

The terms $\partial\bar{r}/\partial\alpha$ and $\partial\bar{r}/\partial\beta$ are simply two rows of the Jacobian matrix evaluated on the surface. After making appropriate substitutions and performing a few matrix manipulations, the expression reduces to:

$$U = \int_{-1}^{1}\int_{-1}^{1} \langle f(\alpha,\beta)\rangle \left\{u_i^*\right\}\ d\alpha d\beta \tag{20}$$

where $\left\{u_i^*\right\}$ is the vector of nodal coordinates for the eight nodes on the face and $f(\alpha,\beta)$ is a row vector of functions of α and β.

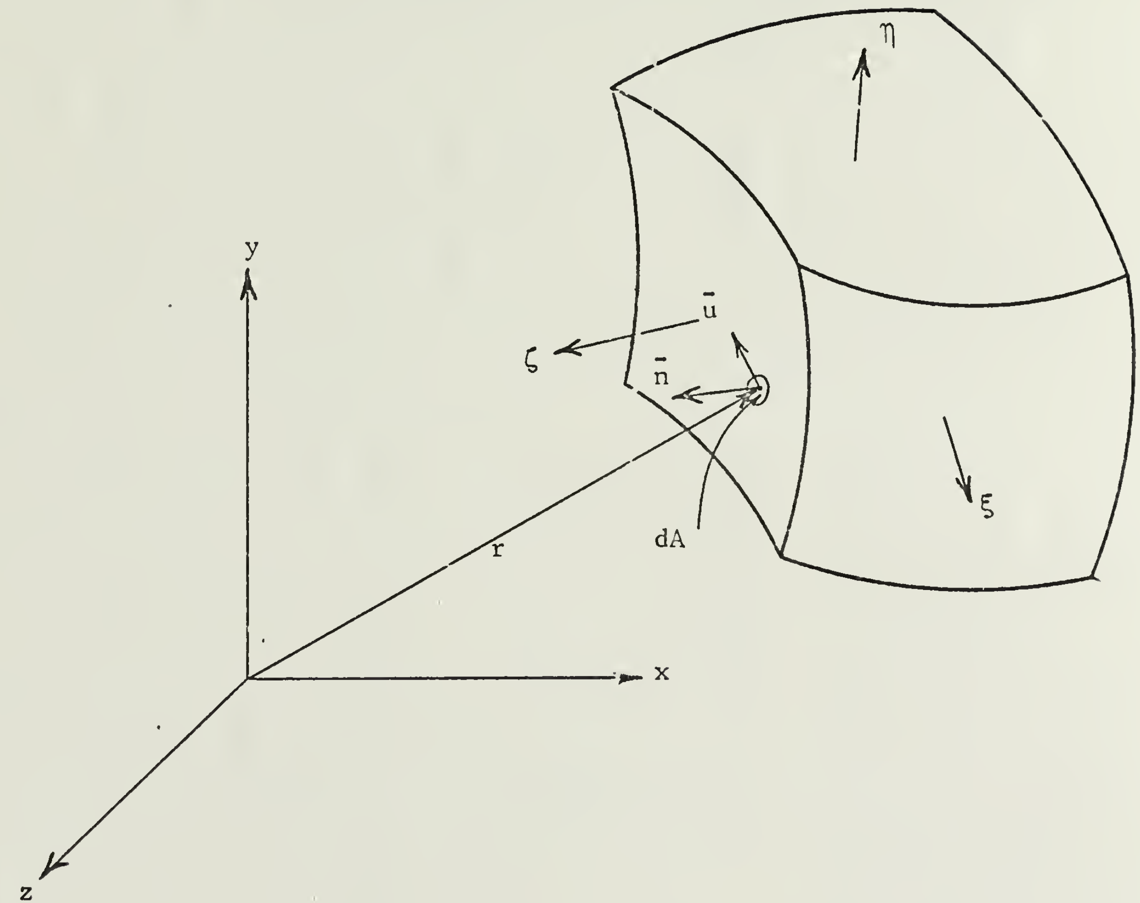


Figure 6. Pressurized Element

The integration is performed by Gaussian quadrature and the resultant vector is exhibited in convenient matrix form. Five Gauss points are used in both QUAMEG and CUMEG.

The Jacobian matrix and shape functions are determined in Subroutine GRAP by using the same function statements as are used in the calculation of the consistent gravity load vector. The face number of the element on which the pressure is acting must be included in the input data. With this information available, a "computed go to" statement is used in the code to determine which variables (ξ, η, ζ) will replace α and β. Referring to Figure 6, the faces are located as described below.

Face	Location	α	β	Face	Location	α	β
1	$\xi = +1$	η	ζ	4	$\xi = -1$	ζ	η
2	$\eta = +1$	ζ	ξ	5	$\eta = -1$	ξ	ζ
3	$\zeta = +1$	ξ	η	6	$\zeta = -1$	η	ξ

H. PLOT OF STRUCTURE MESH

Once the coordinates of all nodes in the mesh have been established, the mesh must be inspected to determine if it is acceptable. Without a plot of the coordinates, this task is nearly impossible. Plotting the mesh by hand is prohibitively time-consuming. Subroutine GRID has been included in QUAMEG and CUMEG to generate an off-line printer plot of the mesh. The parameters required for IBM 360 source library program DRAW are determined in GRID. DRAW is called to perform the actual plotting of the mesh. GRID generates a two-dimensional plot of the mesh on slices at constant values of ζ'. For the example shown in Figure 3, faces 1 and 2 and five slices at constant values of ζ' would be plotted.

The user can specify that the scaling be performed automatically by DRAW. However, that choice could result in the x-scale being not equal to the y-scale. The plot of the mesh would be distorted in that case. The scaling is done in GRID to insure that the scales will be equal and that the smallest possible scale is used.

The plotting space available with DRAW is fifteen inches in the y-direction and nine inches in the x-direction. Recall that the convention for numbering the nodes usually results in having the narrowest side of the structure along the y-axis (section 3.A). Therefore, GRID interchanges the x- and y-coordinates of the structure, for plotting purposes only, to take advantage of the larger plotting space in the y-direction. This further reduces the scale of the plot.

IV. INPUT DATA PREPARATION

This chapter is written as a self-contained unit giving specific information necessary for the preparation of input data cards. The chapter may be removed and used as a guideline for input data preparation after a thorough knowledge of the computer programs has been obtained.

A. VARIABLES

The variables and formulas defined below are included in the code and should be used as an aid for determining the arrangement of nodes within the mesh when preparing input data.

	Variable	Meaning
1.	NEL	Total number of elements.

NEL = (NX)(NY)(NZ), where NX, NY and NZ are the number of elements along the ξ', η', and ζ' directions (21)

2.	NPS	Total number of nodes on any face at constant ζ', where a face is a slice which intersects both corner nodes and mid-side nodes.

NPS = 3(NX)(NY) + 2(NX + NY) + 1 (QUAMEG) (22)
NPS = 5(NX)(NY) + 3(NX + NY) + 1 (CUMEG) (23)

3.	NPM	Total number of mid-side nodes between any two faces.

NPM = (NX + 1)(NY + 1) (QUAMEG) (24)
NPM = 2(NX + 1)(NY + 1) (CUMEG) (25)

4.	NPL	Total number of nodes in one layer, where a layer consists of one face plus the slice intersecting mid-side nodes between that face and an adjacent face.

NPL = NPM + NPS (26)

5.	NUMNP	Total number of nodes in the mesh.

NUMNP = (NZ)(NPL) + NPS (27)

B. DATA CARDS

1. Element Field Data (7I10,2I5)

Number of cards per problem - one

Columns	Variable	Meaning
1 - 10	NX	Number of elements in the ξ'-direction;
11 - 20	NY	Number of elements in the η'-direction;
21 - 30	NZ	Number of elements in the ζ'-direction;
31 - 40	NDX	Number of segments in the ξ'-direction;
41 - 50	NDY	Number of segments in the η'-direction;
51 - 60	NDZ	Number of segments in the ζ'-direction;
61 - 70	NPDP	Total number of nodes for which the coordinates must be specified to define the location of segment end points or other boundary nodes;
71 - 75	KORD	Specifies whether rectangular or cylindrical coordinates are used in the problem. A positive integer indicates cylindrical coordinates and a blank or zero specifies rectangular coordinates;
76 - 80	NCARD	A positive integer indicates that punched cards are desired as output. A blank or zero indicates that cards are not desired.

2. Problem Identification (10A8)

Number of cards per problem - one

Columns	Variable	Meaning
1 - 80	TITLE	Alphameric information to identify the problem.

3. Nodal Point Coordinates (1I10, 3F10.0)

Number of cards per problem - one for each NPDP point

Columns	Variable	Meaning
1 - 10	I	Nodal point number;
11 - 20	COORD(I,1)	x-coordinate (radius);

Columns	Variable	Meaning
21 - 30	COORD(I,2)	y-coordinate (angle);
31 - 40	COORD(I,3)	z-coordinate.

4. Segment Divisions (5I10)

Number of cards per problem - one for each segment

Columns	Variable	Meaning
1 - 10	I	Lowest node number in the segment;
11 - 20	J	Highest node number in the segment;
21 - 30	N1	Segments in the ξ'- and ζ'-directions: Number of elements above the segment (in the positive η'-direction); Segments in the η'-direction: Number of elements within the segment;
31 - 40	M1	Segments in the η'- and ζ'-directions: Number of elements to the left of the segment (in the negative ξ'-direction); Segments in the ξ'-direction: Number of elements within the segment;
41 - 50	L1	Segments in the ξ'- and η'-directions: Number of elements before the segment (in the positive ζ'-direction); Segments in the ζ'-direction: Number of elements within the segment.

5. Load Data (2I10, 2F10.0)

Number of cards per problem - one

Columns	Variable	Meaning
1 - 10	MAP	A positive integer indicates that a plot of the structure mesh is desired. A blank or zero indicates that a plot is not desired;
11 - 20	NEFP	Number of pressurized element faces;
21 - 30	SGZ	Specific weight;
31 - 40	UDP	Uniformly distributed pressure.

6. Pressure Loading Data (2I10)

Number of cards per problem - one for each NEFP

Columns	Variable	Meaning
1 - 10	NELP	Element number on which a pressure exists;
11 - 20	NFACE	Face on which the pressure is applied. (See section 3.G and Figure 6)

7. Plot Title Cards (6A8)

Number of cards per problem - two per plot (NZ + 1 plots)

Columns	Variable	Meaning
1 - 48	ITITLE	Alphameric information to identify the plot;
1 - 48	ITITLE	Alphameric information to identify the programmer and his computer center box number.

8. Stop Trap

The stop trap employed in the programs allows the execution of as many separate problems as may be desired with one submission of the program. The program will continue processing problems until it is informed that NX is less than zero. Therefore, the last card of a complete input data deck should have a negative integer in columns 1 - 10.

V. CONCLUSIONS

The addition of QUAMEG and CUMEG to the tools available in the finite element method of analysis opens new doors for the analyst who uses isoparametric finite elements. Previously, the amount of input data required to analyze a structure of appreciable size was too overwhelming for the analyst to undertake the problem. QUAMEG and CUMEG make it practical for the analyst to solve such a problem. In addition, he can be certain, after an acceptable mesh is generated, that the output data generated by QUAMEG and CUMEG is free of error.

For the example discussed in Appendix A, the number of input cards required by QUAMEG versus the number of output cards generated by QUAMEG for direct input to TRISOP is about six per cent. For the 1 x 8 x 8 mesh of the pinched cylinder problem discussed in Appendix B, this ratio is about 4 per cent.

With a gravity load and pressure load acting on one cubic element, it was found that QUAMEG and CUMEG produced consistent load vectors which were accurate to at least eight decimal places.

QUAMEG and CUMEG considerably reduce the effort required of the analyst to solve a problem by the finite element method, generate acceptable values for nodal point coordinates and compute accurate consistent gravity and pressure load vectors.

VI. RECOMMENDATIONS

A. LOADING CONDITIONS

The capabilities of QUAMEG and CUMEG should be increased by adding subroutines which will generate the consistent load vectors for a thermal gradient in the structure and for an initial displacement condition at one or more nodes.

B. PLOT OF STRUCTURE MESH

It would be preferred to obtain a more complete representation of the structure mesh than is now provided. Two courses of action are recommended:

1. Expand Subroutine GRID in such a way that plots of faces in the η'-ζ' and/or ξ'-ζ' planes are also generated.

2. Replace GRID with a subroutine which would generate a three-dimensional plot of the structure mesh. The feasibility of using the IBM 360 source library program PLOT PACKAGE, which is mentioned in Ref. 12, should be investigated if this method is attempted.

This author feels that the results of recommendation 2 would be preferred to 1, although the coding may be more difficult.

APPENDIX A
GENERATING A MESH WITH QUAMEG

This appendix describes the preparation of input cards and the expected results for a sample problem using QUAMEG. This problem was chosen as an example to demonstrate the capabilities of QUAMEG and to more fully explain the procedures described in Chapter 4.

The structure considered is a cylindrical solid with a star shaped void in the center. This void linearly converges to a point. Figure 7 shows a 45^{o} sector of the structure. It is normally preferable to work with angles in the first quadrant. Therefore, the sector from zero to forty-five degrees will be considered.

The first step in the problem is to decide upon a mesh size. In this case, an 8 x 5 x 4 mesh was chosen. The amount of control to be exercised by the programmer to insure that the structure will be modeled accurately can be determined now. The coordinates of all nodes along the edges of the bounding surfaces of the structure must be available to the system before the remaining nodal coordinates can be determined. Since there are no irregularities on the boundaries between the front and back faces, the coordinates of the nodes on the edges between these two faces can be computed by providing the necessary segment data. Therefore, only the nodes on the front and back faces need be considered.

At this point, two rectangles of constant ζ' are drawn with the proposed mesh superimposed upon them (see Figures 8 and 9). These figures will serve as an aid for numbering the nodes on the

actual structure and for determining the locations of segments. After the nodes on face one are numbered, equation 22 is solved to insure that an error has not been made in the numbering. The result of equation 22 is subtracted from that of equation 27 to determine the number of the first node on the back face. In this way, the programmer need not go through the time-consuming process of numbering all nodes in the mesh. The nodes on the back face are numbered and checked as before. Note that the 8 x 5 mesh is in the $\xi'-\eta'$ plane. Orienting the structure so that the 5 x 4 face of the mesh were in the $\xi'-\eta'$ plane would reduce the half-bandwidth of the system. However, cylindrical coordinates must be used in this problem and it would not be possible to model the structure accurately if it were oriented in that manner. Finally, a picture is drawn of the front and back faces of the structure with the desired mesh superimposed and the nodes numbered (Figures 10 and 11).

The identification of segments will now be determined. The segments will be considered in the same order as they are considered in the program. Referring to Figures 8 and 9, it is seen that the following lines are parallel to the η'-axis: 1 - 11, 137 - 147, 805 - 815, 941 - 951. Now referring to Figures 10 and 11, it is seen that all of these lines are of constant angle with a uniform distribution of elements within the lines. Therefore, these lines will be treated as segments and the coordinates of the nodes listed above will be included in the input data.

By this method of specifying lines, the author does not intend to imply that the nodes along the line are numbered consecutively, starting with the lower number. For example, the nodes along line

60 - 111 in Figure 8 are: 60, 67, 77, 84, 94, 101, and 111. This method of specifying lines merely lists the numbers of nodes at the ends of a line.

In order to control the departure angles of lines 79 - 77, 96 - 94, and 113 - 111, it is necessary to force line 52 - 62 nearer the x-axis than would result if an even distribution of nodes along line 1 - 137 were allowed. Therefore, an appropriate location for node 52 is chosen, the coordinates of nodes 52 and 62 are included in the input data, and line 52 - 62 is treated as a segment. Note that this line is of neither constant angle nor constant radius. However, since the line is not on the edge of a bounding surface, the fact that it will be curved will not affect the solution adversely. Similarly, the coordinates of nodes 856 and 866 will be included in the input data and that line will be a segment.

The same rules govern the selection of segments in the ξ'-direction. Referring to Figures 8 and 9, it is seen that the coordinates of nodes along lines 1 - 137, 11 - 147, 805 - 941, and 815 - 951 must be made available to the computer. In Figures 10 and 11, line 1 - 137 is a line of constant radius; however, the element divisions along this line are not uniform. Two segments must be specified: lines 1 - 52 and 52 - 137. The coordinates of the end nodes on these segments have been specified already and need not be repeated. Line 11 - 62 is of constant radius and is therefore one segment. Again, the coordinates of these nodes have been specified previously. Line 62 - 113 is of constant angle, meeting the criterion for a segment. The coordinates of node 113 must be specified. Line 113 - 147 is of neither constant angle nor constant radius. Therefore, it cannot

be treated as a segment and the coordinates of all nodes along this line must be included in the input data. In order to further control the departure angle of the lines mentioned above, line 60 - 145 is also included as a segment. The coordinates of the end nodes of this segment have been computed in the problem already and do not need to be included in the input data. As with line 52 - 62, the exact shape of line 60 - 145 is not important. Similarly, on the back face, lines 805 - 856, 856 - 941, and 864 - 949 are treated as segments. Although the nodes along line 815 - 951 are at the origin, care must be taken to insure that the coordinates of these nodes are specified at appropriate angles in such a way that this point will have the same "shape" as the front face. The radial component of these coordinates is zero. This care is necessary because all computations in the problem are made with cylindrical coordinates. Were the coordinates of the nodes along this line specified at a radius of zero and an angle of zero, the mesh would be distorted. The following lines can also be treated as segments: 815 - 866 and 866 - 917. The coordinates of nodes 923, 934, and 940 must be included in the input data for the same reason as was given for nodes 119, 130, and 136.

To complete the process of specifying the coordinates of nodes along the structure edges, the following lines will be treated as segments in the ζ'-direction: 1 - 805, 52 - 856 137 - 941, 60 - 864, 77 - 881, 94 - 898, 11 - 915, 11 - 815, 62 - 866, 113 - 917, and 130 - 934.

The procedure for determining the values of N1, M1, and L1 for the segment division data will not be discussed here. It is

felt that the explanation in Chapter 4, combined with the listing of the input cards for this problem, is sufficiently clear.

The card containing the element field data is prepared after the above steps have been accomplished. Loading information and the request for punched cards should not be included in the input data until it has been determined that the mesh will be satisfactory.

Approximately 45 seconds of computer time were required to solve this problem with QUAMEG. That time includes the 31 seconds required to compile the program. Figures 12, 13, 14, 15, and 16 are the plots of the structure mesh that was generated by QUAMEG. Table I is the listing of the required input cards described above. Figure 17 is the result of a 5 x 5 x 3 mesh generated with CUMEG for the same structure. Intermediate slices and the back surface are not shown.

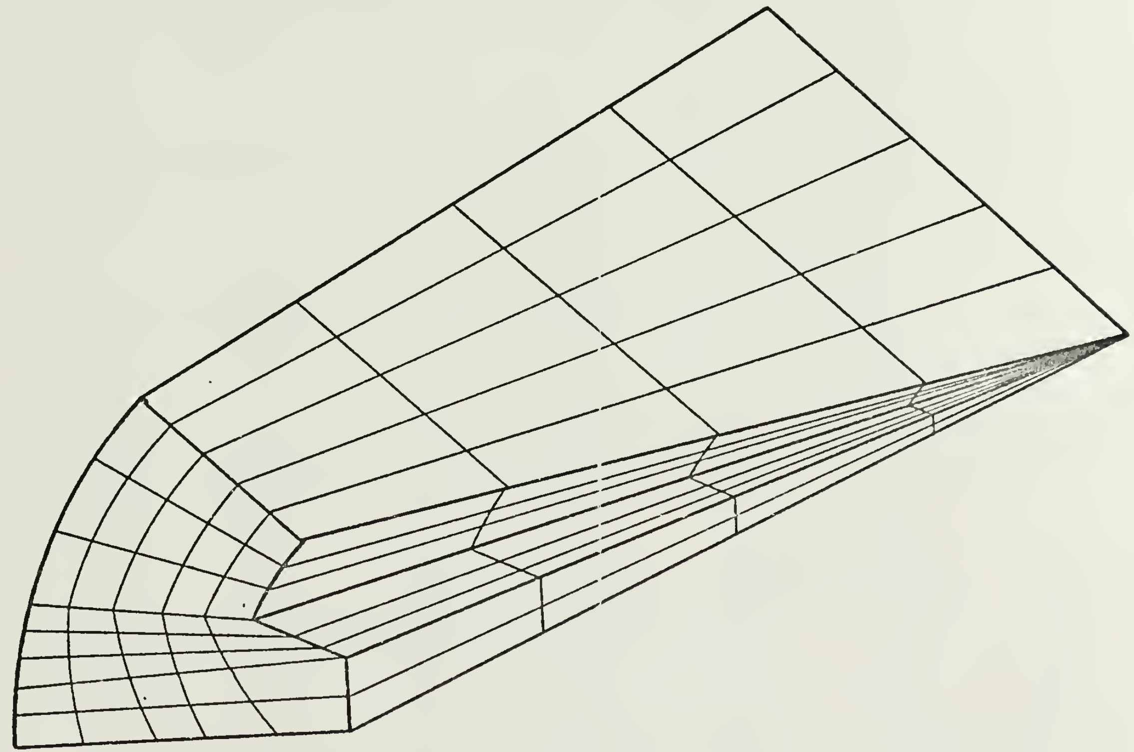


Figure 7. Problem Structure

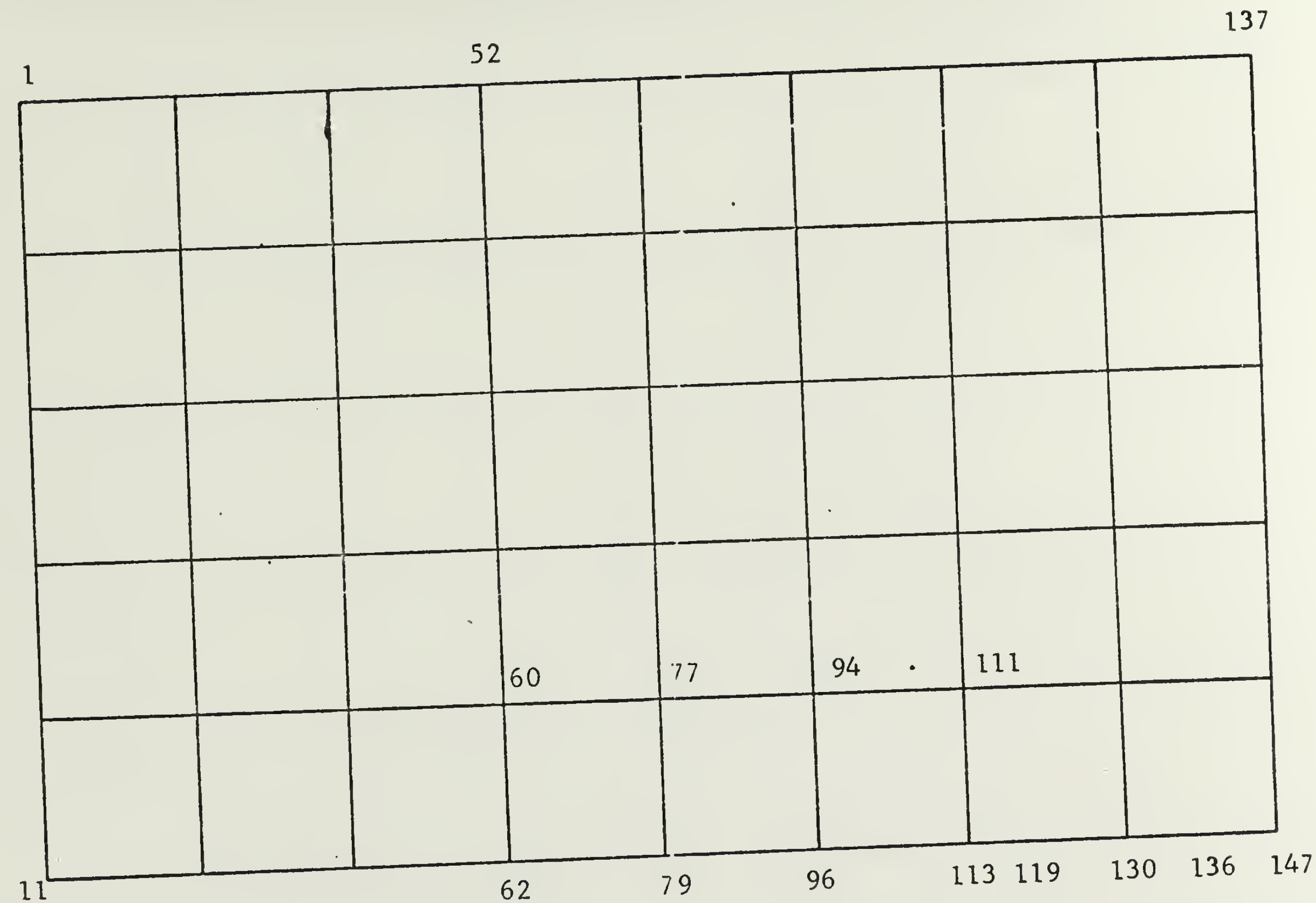


Figure 8. Front Face

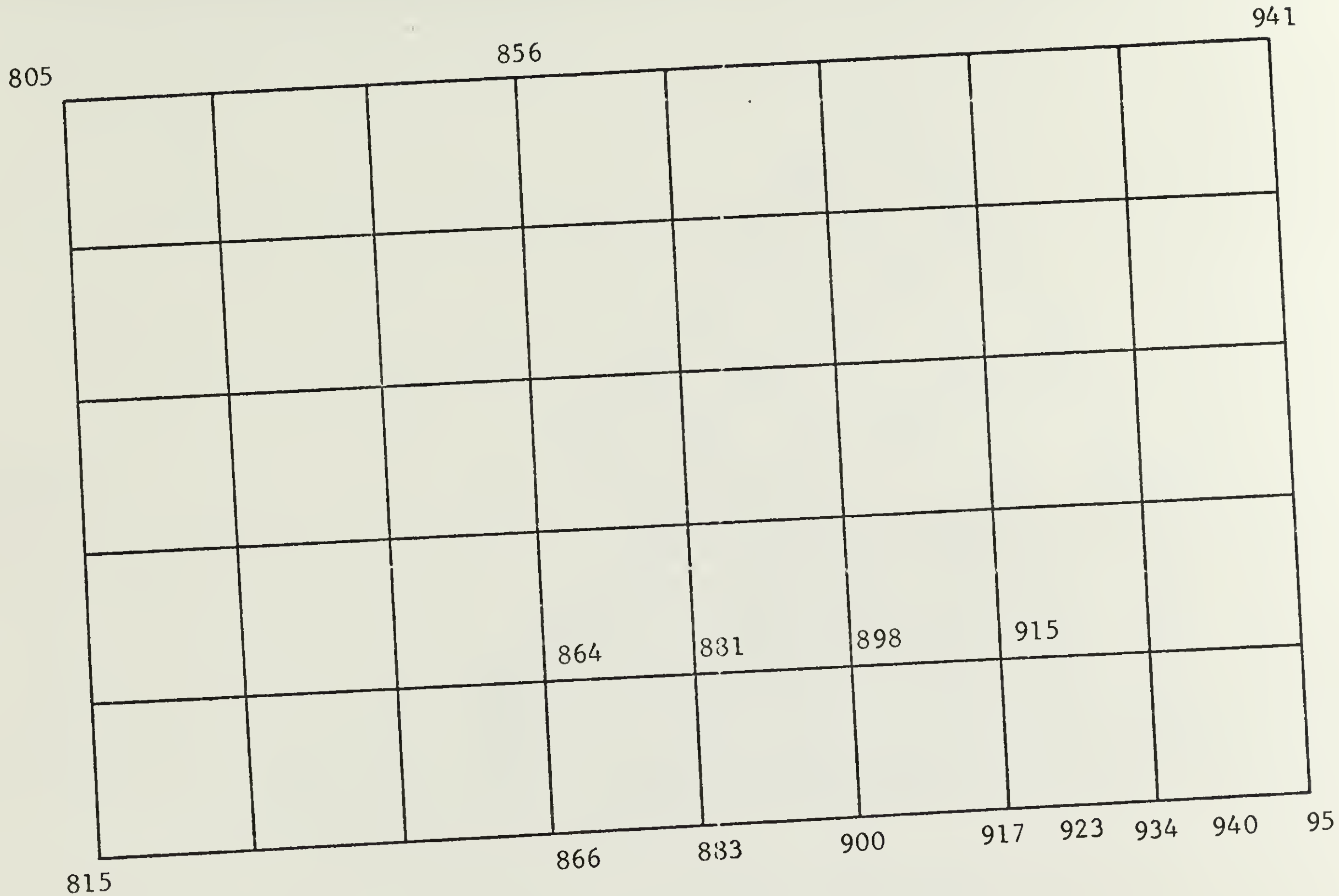


Figure 9. Back Face

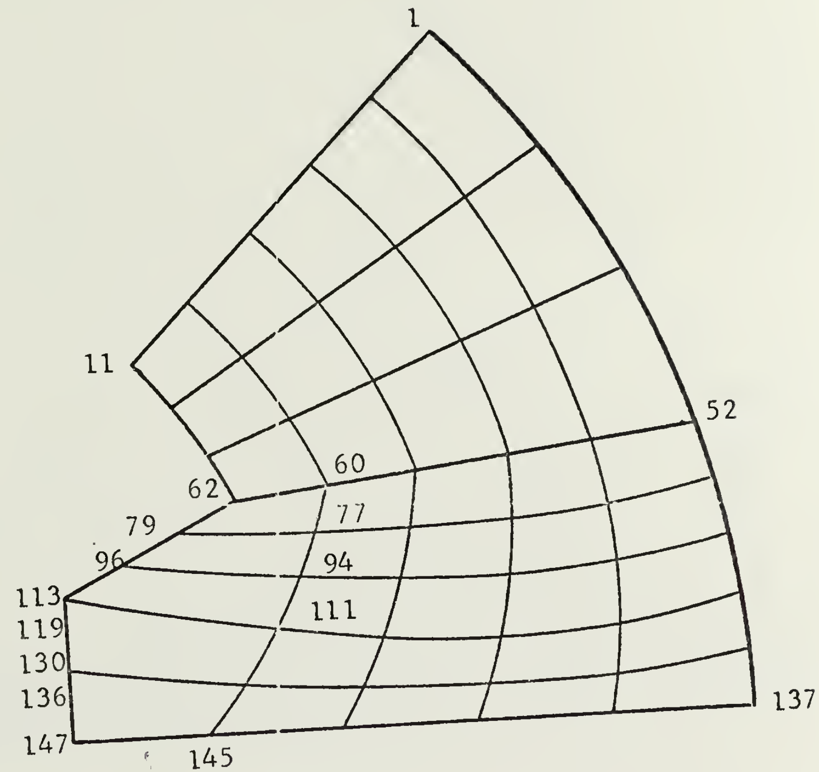


Figure 10. Front Face of Structure

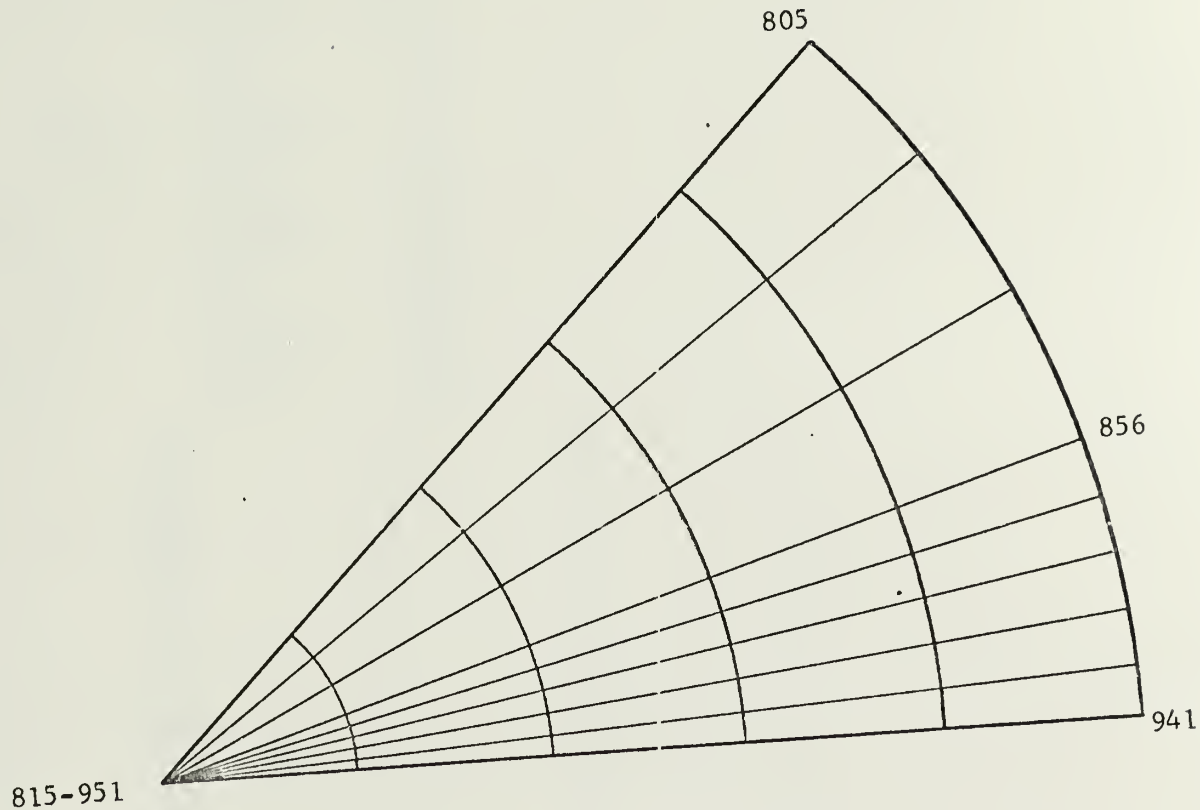


Figure 11. Back Face of Structure

Table I
Listing of Input Data Cards

```
   1        8        5        4       10        6       12      201
                  ROCKET MOTOR PROPELLENT MESH (8X5X4)
       15.0       45.0       5.0
      112.677     45.0       5.0
      622.677     26.565     5.0
     1131.677     26.565     5.0
     1471.5        0.0       5.0
      525.0       17.0       5.0
     1375.00       0.0       5.0
     1191.602     20.556     5.0
     1301.54615   14.036     5.0
     1361.51168    7.125     5.0
     8150.0       45.0       0.00
     8660.0       26.565     0.00
     9170.0       26.565     0.00
     8055.00      45.0       0.00
     8565.0       17.0       0.00
     9415.0        0.0       0.00
     9230.0       20.556     0.00
     9340.0       14.036     0.00
     9400.0        7.125     0.00
     9510.0        0.0       0.00
        1        11        5        0        0
       52        62        5        3        0
      137       147        5        8        0
      805       815        5        0        4
      856       866        5        3        4
      941       951        5        8        4
        1        52        0        3        0
       52       137        0        5        0
       60       145        4        5        0
       11        62        5        3        0
       62       113        5        3        4
      805       856        0        3        4
      856       941        0        5        4
      815       866        5        3        4
      866       917        5        3        4
      864       949        4        5        4
        1       805        0        0        4
       52       856        0        3        4
      137       941        0        8        4
       11       815        5        0        4
       62       866        5        3        4
      113       917        5        6        4
      130       934        5        7        4
      147       951        5        8        4
       60       864        4        3        4
       77       881        4        4        4
```

```
    94          898          4          5          4
   111          915          4          6          4
     1                0.0          0.0
      MESH II   FACE 1
     HANSON   BOX 15
      MESH II   FACE 2
     HANSON   BOX 15
      MESH II   FACE 3
     HANSON   BOX 15
      MESH II   FACE 4
     HANSON   BOX 15
      MESH II   FACE 5
     HANSON   BOX 15
-111111
```

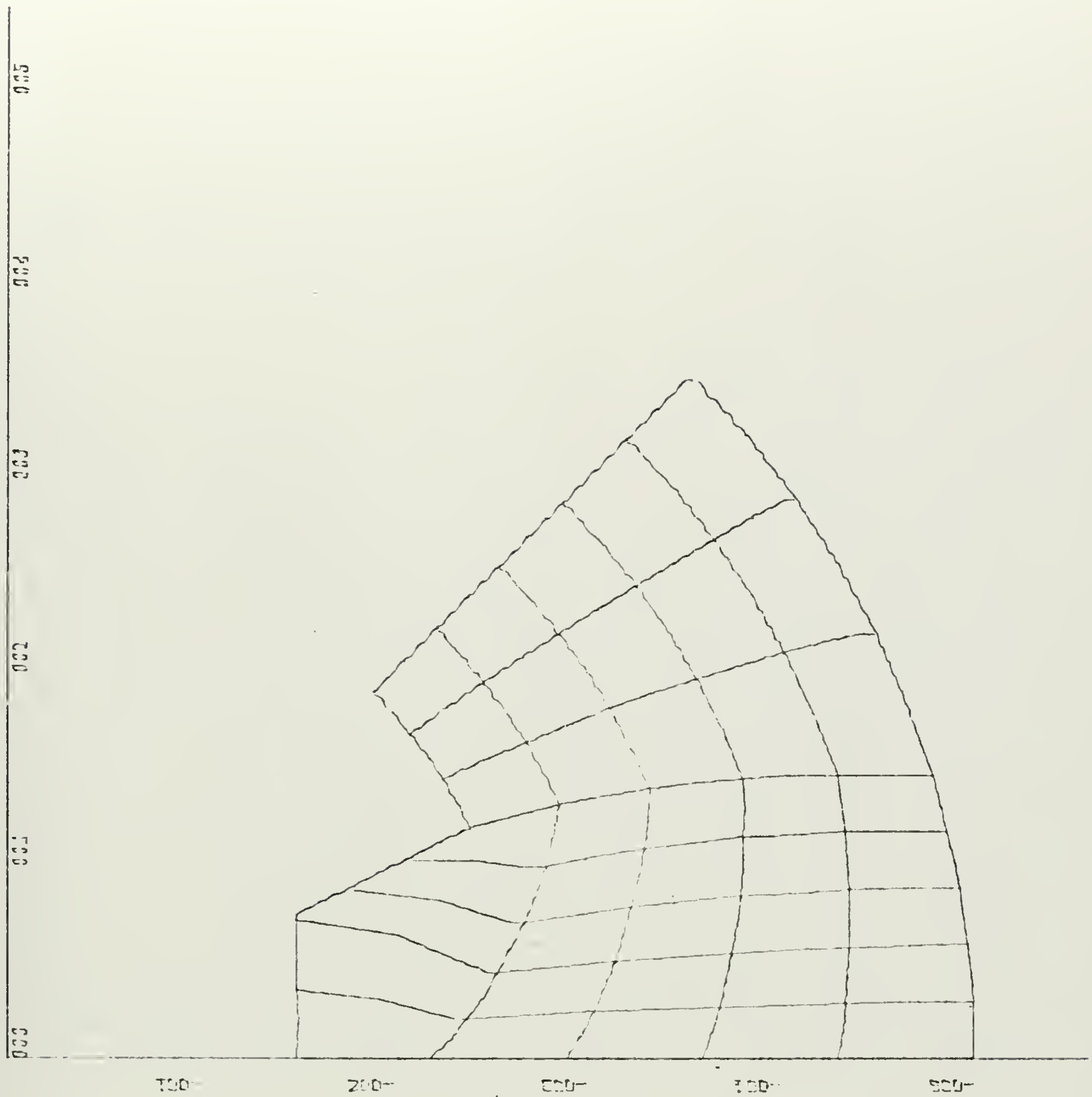


Figure 12. Generated Mesh For Front Face

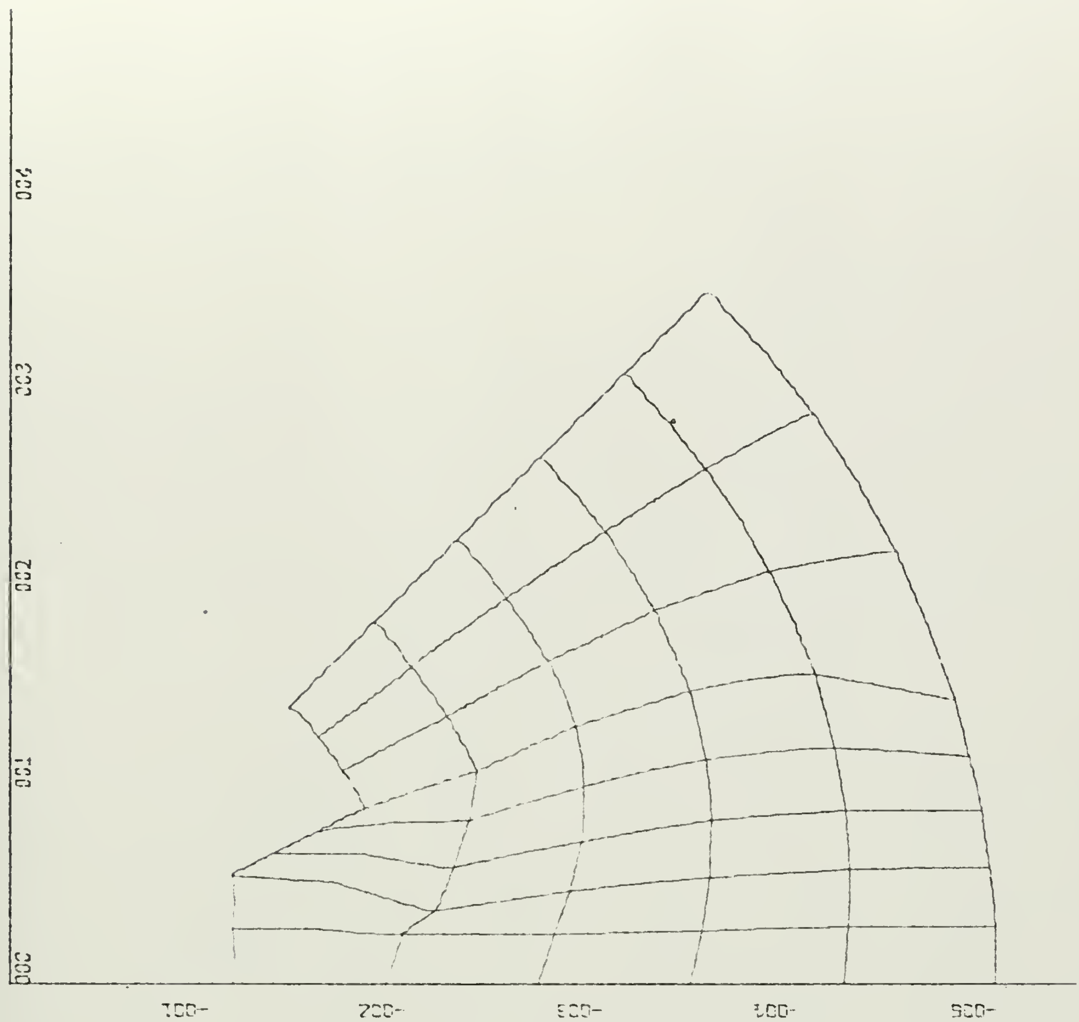


Figure 13. Generated Mesh For First Intermediate Slice

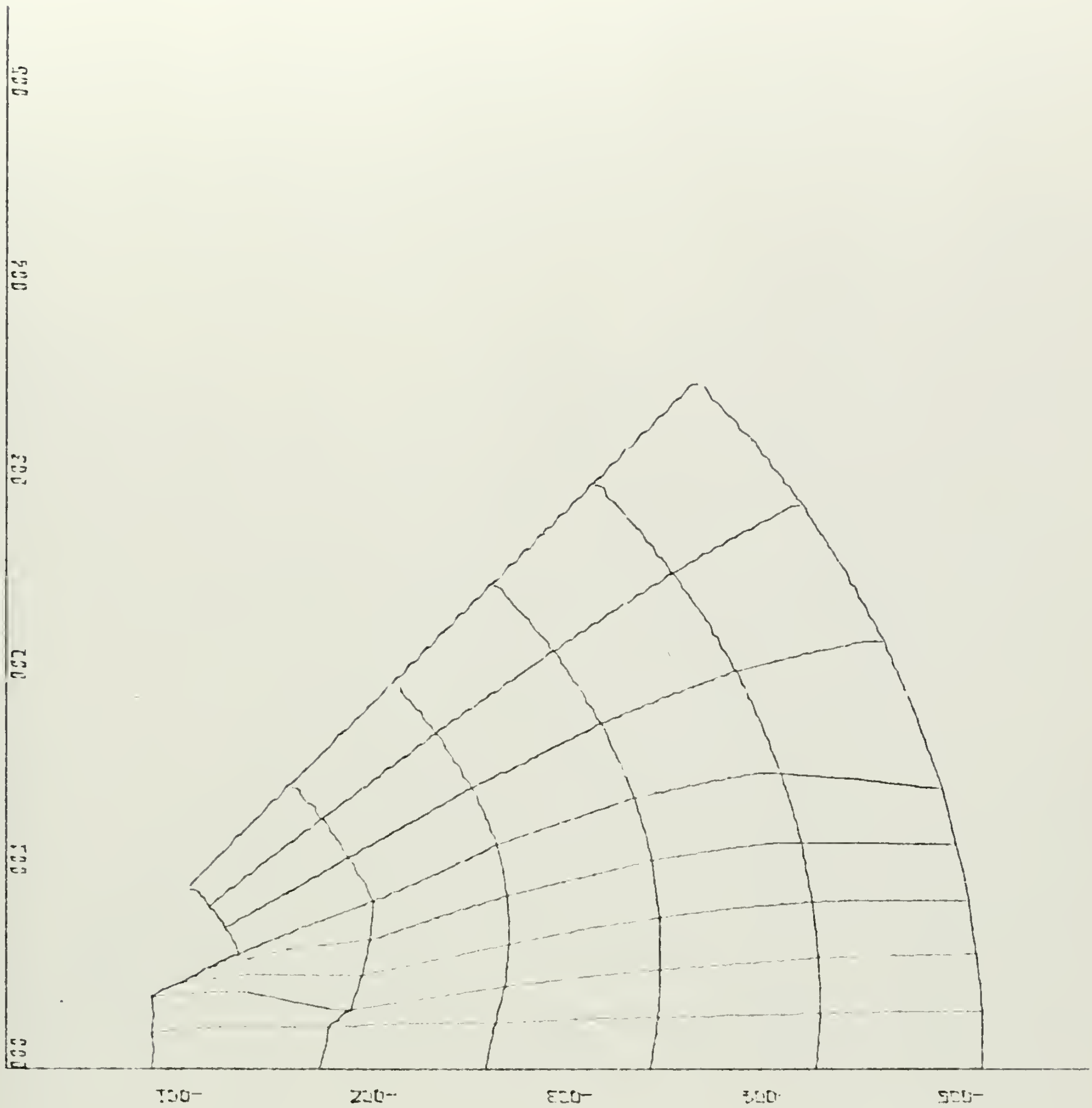


Figure 14. Generated Mesh For Second Intermediate Slice

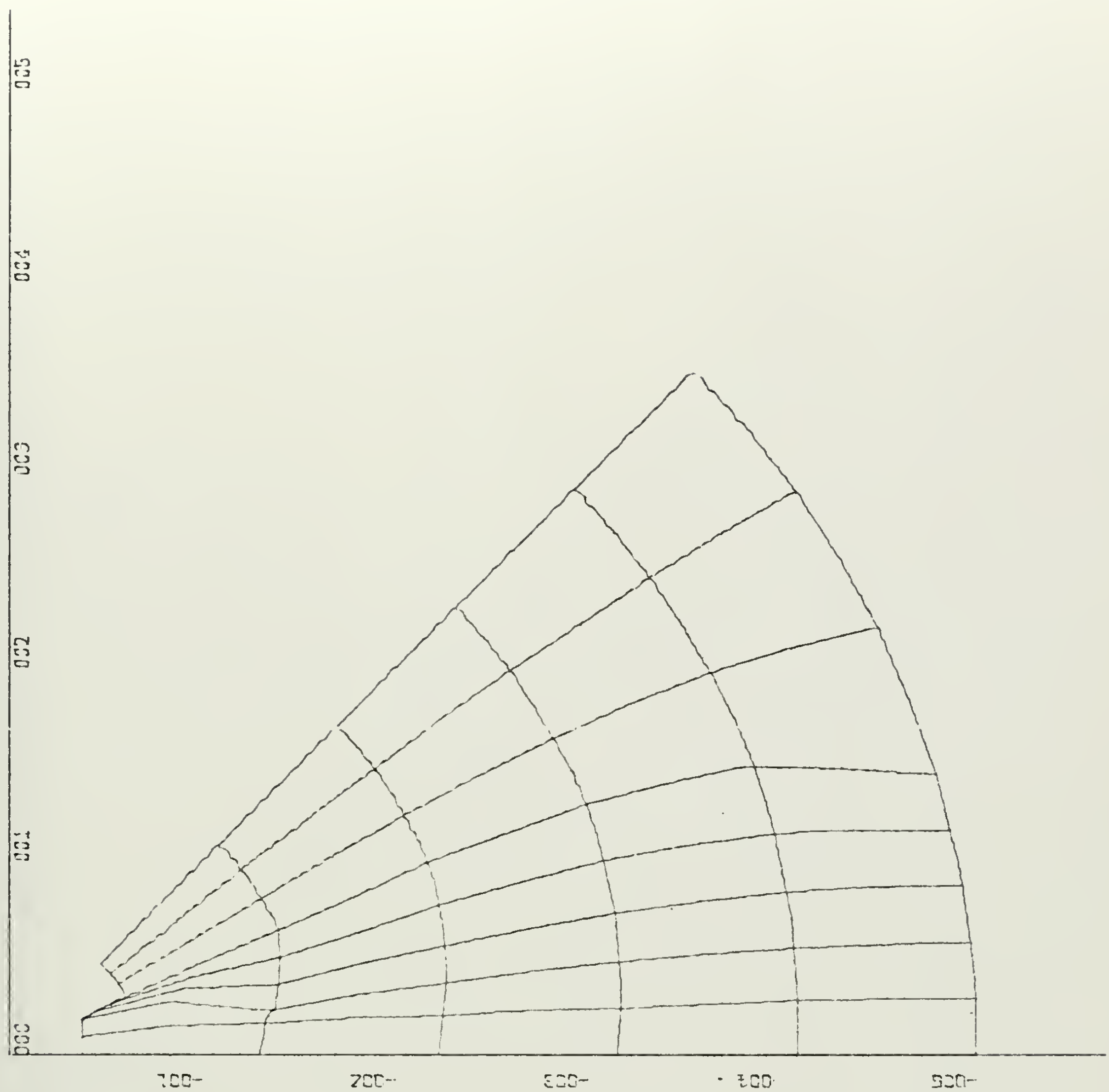


Figure 15. Generated Mesh For Third Intermediate Slice

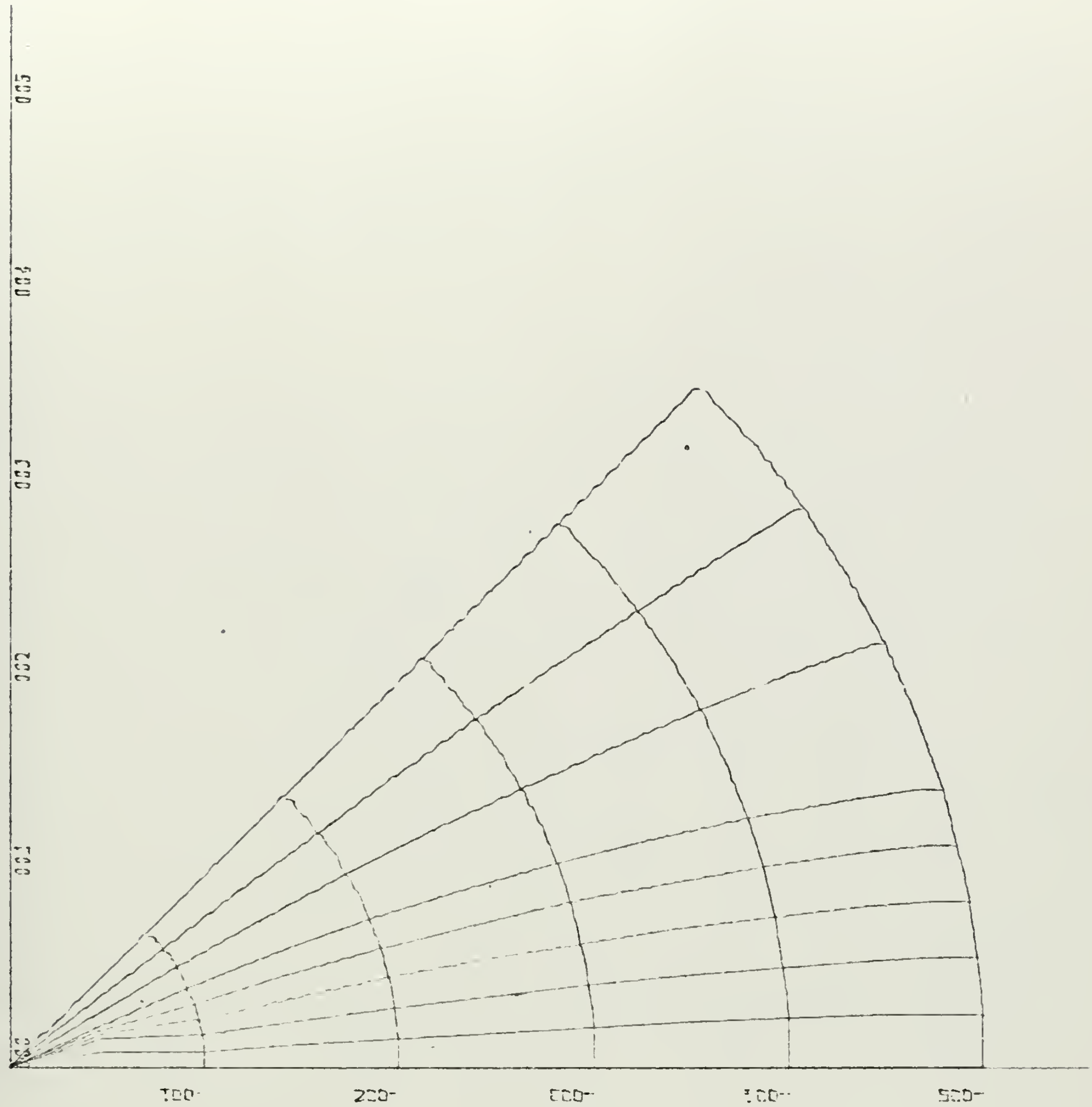


Figure 16. Generated Mesh For Back Face

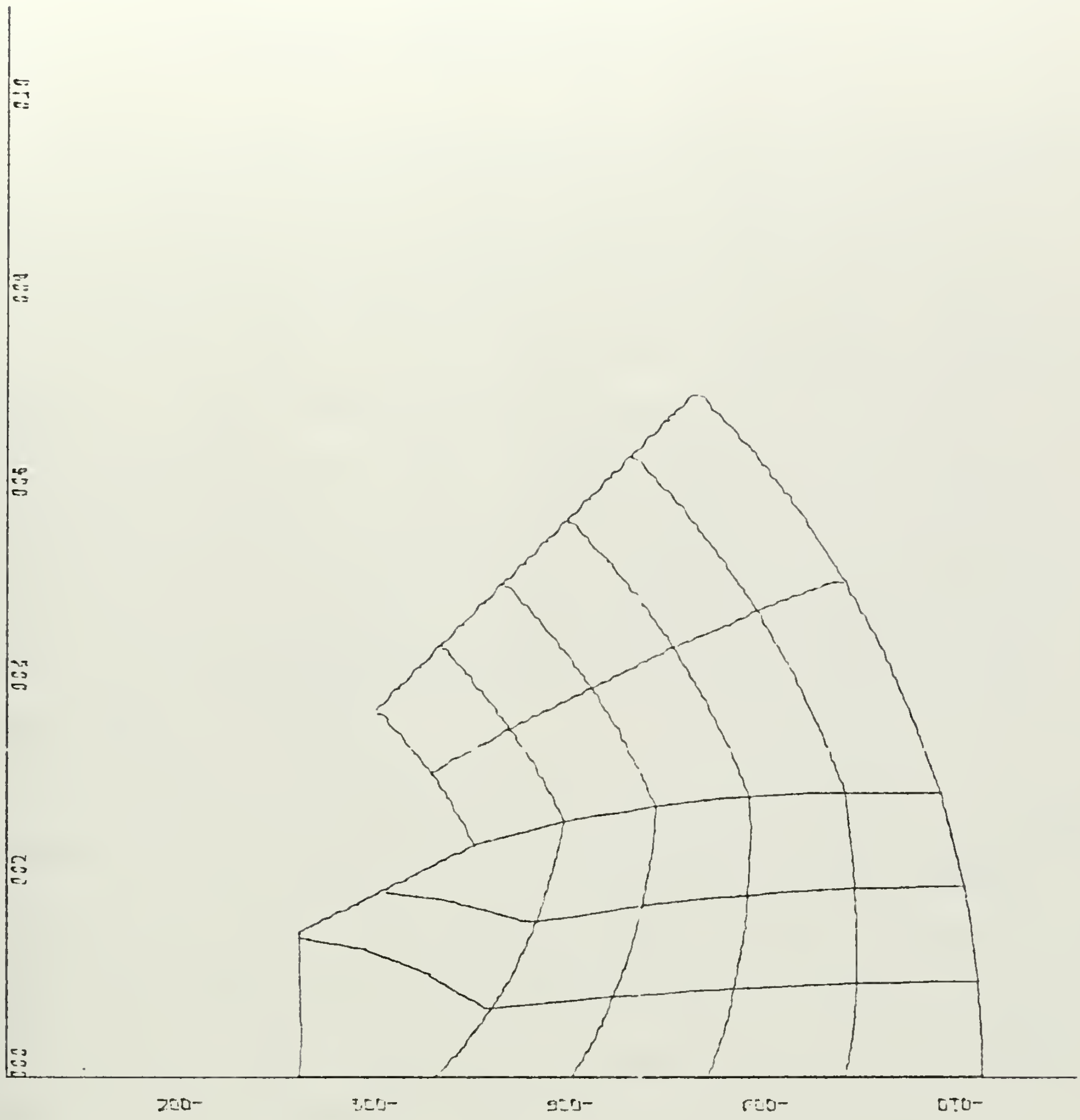


Figure 17. Generated Mesh For Front Face

APPENDIX B
A CLASSICAL PROBLEM SOLVED WITH TRISOP

TRISOP is a relatively new computer system which uses isoparametric finite elements to obtain stresses, strains, and displacements of three-dimensional structures that are subjected to static loads.

Although a number of test problems have been conducted with TRISOP, the input preparation was too voluminous to contemplate a full evaluation of the system. Without QUAMEG, the difficulties discussed in Chapter 1 limited the size and number of problems that could be solved. In order to evaluate TRISOP as a complete system, a classical problem was selected and a convergence study was made, using QUAMEG and TRISOP.

The problem selected was that of the pinched, thin-shell cylinder. Obviously, this problem should be solved using thin-shell elements for maximum efficiency. The purpose of this test was to determine if the three-dimensional solution would converge to the same results as are reported in Ref. 3 and also to determine how well thin-shell action is captured by TRISOP. The results reported by Cantin [3] are included in Table II with the results obtained in the test. The problem is described in Figure 18.

Conclusions:

Although the results of this convergence study indicate a tendency to approach the same results obtained by thin-shell theory, acceptable results can be obtained only with a very fine mesh.

TRISOP can be used to solve problems for which there are no closed-form or accurate numerical methods of solution available. However, the user must be prepared to pay the price of long computational times, for some problems.

Table II
Displacement at the applied load for a pinched cylinder.

Bogner et al.

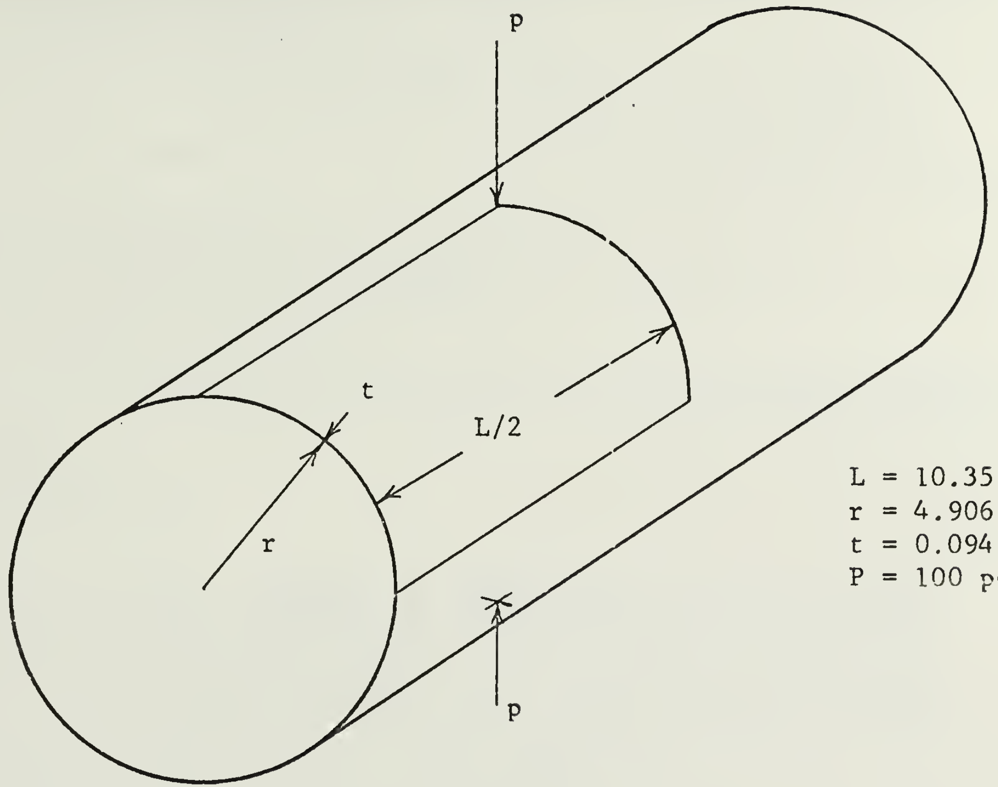
No. of Eq.	Mesh	Displace., in.
72	1 x 2	-0.0802
96	1 x 3	-0.1026
120	1 x 4	-0.1087
108	2 x 2	-0.0808
144	2 x 3	-0.1036
180	2 x 4	-0.1098

Cantin

No. of Eq.	Mesh	Displac., in.
36	1 x 2	-0.0921
48	1 x 3	-0.1072
60	1 x 4	-0.1099
54	2 x 2	-0.0931
72	2 x 3	-0.1085
90	2 x 4	-0.1113
150	4 x 4	-0.1126
294	6 x 6	-0.1137
486	8 x 8	-0.1139
726	10 x 10	-0.1139

Hanson

No. of Eq.	Mesh	Displac., in.
153	1 x 2 x 2	-0.00177
243	2 x 2 x 2	-0.00178
423	2 x 2 x 4	-0.00196
465	1 x 4 x 4	-0.00375
945	1 x 6 x 6	-0.03002
1593	1 x 8 x 8	-0.08702



L = 10.35 inches
r = 4.906 inches
t = 0.094 inches
P = 100 pounds

Figure 18. Pinched Cylinder

APPENDIX C
COMPUTER LISTING (QUAMEG)

```
C
C      ************************************************************
C      *                                                          *
C      *  MESH GENERATING PROGRAM FOR TRISOP                      *
C      *  USING QUADRATIC ELEMENTS                                *
C      *  CODED BY D. E. HANSON, FEBRUARY 1971                    *
C      *  FRONT FACE IS IN XY PLANE, LEFT FACE IS IN YZ PLANE     *
C      *  NODAL POINT ONE IS AT UPPER LEFT CORNER OF FRONT FACE   *
C      *                                                          *
C      ************************************************************
C
       IMPLICIT REAL*8(A-H,O-Z)                                         QMG00010
       DIMENSION GLV(999),PLM(999,3),NELC(8)                            QMG00020
       DIMENSION NCON(200,21),KODE(999),NC(6,8),TITLE(10)               QMG00030
       COMMON /NB1/ NPEL,NEL,NFACE,NUMNP                                QMG00040
       COMMON /B2/ COORD(999,3),COREL(20,3)                             QMG00050
       COMMON /B3/ AK(20),AK1(20)                                       QMG00060
       COMMON /B4/ AK2(8,3),AK3(8,3)                                    QMG00070
       DATA NC/13,13,1,3,17,15,9,14,2,10,18,14,1,15,3,15,19,13,8,10,4,16,QMG00080
      112,20,7,3,5,17,7,19,12,2,6,11,6,18,19,1,7,5,5,17,20,9,8,4,11,16/ QMG00090
 1000 FORMAT('1')                                                       QMG00100
 1100 FORMAT (7I10,2I5)                                                 QMG00110
 1200 FORMAT (5I10)                                                     QMG00120
 1300 FORMAT(1I10,3F10.0)                                               QMG00130
 1400 FORMAT(1X,20I3,1I5)                                               QMG00140
 1500 FORMAT(///,                                                       QMG00150
      115X,'NEL........TOTAL NUMBER OF ELEMENTS.............'I5/,       QMG00160
      215X,'NUMNP......TOTAL NUMBER OF NODAL POINTS.........'I5/,       QMG00170
      315X,'NEQ........TOTAL NUMBER OF EQUATIONS............'I5/,       QMG00180
      415X,'NBAND......HALF-BAND WIDTH OF THE SYSTEM........'I5/,       QMG00190
      515X,'NN.........NUMBER OF BLOCKS PER COLUMN..........'I5/,       QMG00200
      615X,'MM.........NUMBER OF BLOCKS PER ROW.............'I5/,       QMG00210
      A15X,'TIME.......ESTIMATED SOLVE TIME (IN MINUTES)....'I5/,       QMG00220
      715X,'FT07F001...SPACE=(480,(',I5,',60))'/,                       QMG00230
      815X,'FT08F001...SPACE=(480,(',I5,',60))'/,                       QMG00240
      915X,'FT11F001...SPACE=(',I5,',(1,1))'/,                          QMG00250
      115X,'FT12F001...SPACE=(48,(',I5,',48))'/,                        QMG00260
     1215X,'FT13F001...SPACE=(48,(',I5,',48))'//)                       QMG00270
 1600 FORMAT(//,                                                        QMG00280
      136X,'CONNECTIVITY MATRIX'//,                                     QMG00290
      211X,'  EL   A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q  R QMG00300
      3S  T    NMAT'//)                                                 QMG00310
 1700 FORMAT(11X,I4,2X,20I3,I5)                                         QMG00320
 1800 FORMAT(6X,I10,3F15.5)                                             QMG00330
 1900 FORMAT(2I10,2F10.0)                                               QMG00340
 2000 FORMAT(///35X,'COORDINATES'//,14X,'NP', 9X,'X',14X,'Y',14X,'Z'/)  QMG00350
 2100 FORMAT(///,20X,'GRAVITY LOAD',//,14X,'NP',11X,'Z'/)               QMG00360
 2200 FORMAT(///22X,'PRESSURE INTENSITY =',F5.2,//27X,'EL',5X,'FACE'/)  QMG00370
```

```
 2300 FORMAT(6X,I10,F20.12)                                                 QMG00380
 2400 FORMAT(I10,20X,F20.12)                                                QMG00390
 2500 FORMAT(///,30X,'PRESSURE LOADS',//,14X,'NP', 9X,'X',14X,'Y',14X,      QMG00400
     1'Z',//)                                                               QMG00410
 2600 FORMAT(I10,3F20.12)                                                   QMG00420
 2700 FORMAT(2I10)                                                          QMG00430
 2800 FORMAT(10A8)                                                          QMG00440
 2900 FORMAT(24X,I5,7X,I1)                                                  QMG00450
   10 CONTINUE                                                              QMG00460
      READ(5,1100) NX,NY,NZ,NDX,NDY,NDZ,NPDP,KORD,NCARD                     QMG00470
      IF(NX.LT.0) STOP                                                      QMG00480
      READ(5,2800) TITLE                                                    QMG00490
      WRITE(6,2800) TITLE                                                   QMG00500
      PI=3.1415926536D0/180.0D0                                             QMG00510
      ZRO=0.0D0                                                             QMG00520
      ORF=1.8D0                                                             QMG00530
      FP=4.0D0                                                              QMG00540
      NEL = NX*NY*NZ                                                        QMG00550
      NX1=NX-1                                                              QMG00560
      NZ1=NZ-1                                                              QMG00570
      NY1 = NY+1                                                            QMG00580
      NY2 = 2*NY                                                            QMG00590
      NY3 = 3*NY                                                            QMG00600
      NY4=NY2+1                                                             QMG00610
      NY5=NY3+2                                                             QMG00620
      NY6=NY-1                                                              QMG00630
      NY7=NY2-1                                                             QMG00640
      NPS=NX*NY5+NY4                                                        QMG00650
      NPM=NY1*(NX+1)                                                        QMG00660
      NPL = NPS + NPM                                                       QMG00670
      NUMNP = NZ*NPL + NPS                                                  QMG00680
C
C     COMPUTE ELEMENT CONNECTIVITY
C
      DO 15 IN=1,NZ                                                         QMG00690
      IZ = IN - 1                                                           QMG00700
      DO 15 IR=1,NX                                                         QMG00710
      IM = IR - 1                                                           QMG00720
      NR = NX - IM                                                          QMG00730
      IC = IM*NY                                                            QMG00740
      I1 = IM*NY5                                                           QMG00750
      I2 = I1 + 1                                                           QMG00760
      I3 = I2 + 1                                                           QMG00770
      I4 = I1 + NY4                                                         QMG00780
      I5 = I4 + 1                                                           QMG00790
      I6 = I5 + NY                                                          QMG00800
      I7 = I6 + 1                                                           QMG00810
      I8 = I7 + 1                                                           QMG00820
```

```
      I9 =  I1+NR*NY5+NY4+IM*NY1                                        QMG00830
      I10 =  I9 + 1                                                     QMG00840
      I11 =  I10 + NY                                                   QMG00850
      I12 =  I11 + 1                                                    QMG00860
      I13 =  I9 + NY1*(NR+1) + IM*NY5                                   QMG00870
      I14 =  I13 + 1                                                    QMG00880
      I15 =  I14 + 1                                                    QMG00890
      I16 =  I13 + NY4                                                  QMG00900
      I17 =  I16 + 1                                                    QMG00910
      I18 =  I17 + NY                                                   QMG00920
      I19 =  I18 + 1                                                    QMG00930
      I20 =  I19 + 1                                                    QMG00940
      DO 15 I=1,NY                                                      QMG00950
      IL = IZ*NX*NY + IC + I                                            QMG00960
      IT = 2*I - 1                                                      QMG00970
      NCON(IL, 1) = I6  + IT + IZ*NPL                                   QMG00980
      NCON(IL, 2) = I4  + I  + IZ*NPL                                   QMG00990
      NCON(IL, 3) = I1  + IT + IZ*NPL                                   QMG01000
      NCON(IL, 4) = I2  + IT + IZ*NPL                                   QMG01010
      NCON(IL, 5) = I3  + IT + IZ*NPL                                   QMG01020
      NCON(IL, 6) = I5  + I  + IZ*NPL                                   QMG01030
      NCON(IL, 7) = I8  + IT + IZ*NPL                                   QMG01040
      NCON(IL, 8) = I7  + IT + IZ*NPL                                   QMG01050
      NCON(IL, 9) = I11 + I  + IZ*NPL                                   QMG01060
      NCON(IL,10) = I9  + I  + IZ*NPL                                   QMG01070
      NCON(IL,11) = I10 + I  + IZ*NPL                                   QMG01080
      NCON(IL,12) = I12 + I  + IZ*NPL                                   QMG01090
      NCON(IL,13) = I18 + IT + IZ*NPL                                   QMG01100
      NCON(IL,14) = I16 + I  + IZ*NPL                                   QMG01110
      NCON(IL,15) = I13 + IT + IZ*NPL                                   QMG01120
      NCON(IL,16) = I14 + IT + IZ*NPL                                   QMG01130
      NCON(IL,17) = I15 + IT + IZ*NPL                                   QMG01140
      NCON(IL,18) = I17 + I  + IZ*NPL                                   QMG01150
      NCON(IL,19) = I20 + IT + IZ*NPL                                   QMG01160
      NCON(IL,20) = I19 + IT + IZ*NPL                                   QMG01170
      NCON(IL,21) = 1                                                   QMG01180
   15 CONTINUE                                                          QMG01190
C
C     COMPUTE BANDWIDTH, SOLUTION TIME, AND SPACE REQUIREMENTS
C
      NBAND=0                                                           QMG01200
      NS=60                                                             QMG01210
      SN=NS                                                             QMG01220
      NDF=3                                                             QMG01230
      NPEL=20                                                           QMG01240
      NST=NPEL*NDF                                                      QMG01250
      NEQ=NUMNP*NDF                                                     QMG01260
      NN=(NEQ+NS-1)/NS                                                  QMG01270
```

```
      NT=(NEQ+49)/50                                                    QMGC1280
      DO 20 I=1,NEL                                                     QMG01290
      DO 20 J=1,19                                                      QMG01300
      JP=J+1                                                            QMG01310
      DO 20 K=JP,20                                                     QMG01320
   20 NBAND=MAX0(NBAND,IABS(NCON(I,J)-NCON(I,K)))                       QMG01330
      NBAND=(NBAND+1)*NDF                                               QMG01340
      MM=(NBAND+2*(NS-1))/NS                                            QMG01350
      MT=(NBAND+98)/50                                                  QMG01360
      TK=(1.857D0*NT*MT**2+0.102D0*MT*NT**2-0.322D0*NT**2+4.682D0*NT*MT QMG01370
     1-5.23D0*MT**2)/60.0D0                                             QMG01380
      FT=(50.0D0/SN)**2                                                 QMG01390
      KT=TK*FT+0.5D0                                                    QMG01400
      K7=NN*(MM+1)*60                                                   QMG01410
      K8=NEL*60                                                         QMG01420
      K11=NEQ*8                                                         QMG01430
      K12=NEL*20                                                        QMG01440
      WRITE(6,1500) NEL,NUMNP,NEQ,NBAND,NN,MM,KT,K7,K8,K11,K12,K12      QMG01450
      WRITE(6,1600)                                                     QMG01460
      DO 40 I=1,NEL                                                     QMG01470
      IF(NCARD.EQ.0) GO TO 30                                           QMG01480
      WRITE(7,1400)(NCON(I,J), J = 1,21)                                QMG01490
   30 WRITE(6,1700) I,(NCON(I,J),J=1,21)                                QMG01500
   40 CONTINUE                                                          QMG01510
      DO 50 I=1,NUMNP                                                   QMG01520
      GLV(I)=ZRO                                                        QMG01530
      KODE(I) =ZRO                                                      QMG01540
      DO 50 J=1,NDF                                                     QMG01550
      COORD(I,J)=ZRO                                                    QMG01560
   50 PLM(I,J)=ZRO                                                      QMG01570
      DO 60 J=1,NPDP                                                    QMG01580
      READ(5,1300) I,(COORD(I,IR),IR=1,3)                               QMG01590
   60 KODE(I)=1                                                         QMG01600
C
C     COMPUTE COORDINATES OF EDGES
C
      IF(NDY.EQ.0) GO TO 80                                             QMG01610
      DO 70 LA=1,NDY                                                    QMG01620
      READ(5,1200) I,J,N1,M1,L1                                         QMG01630
      ND=2*N1                                                           QMG01640
      DX=(COORD(J,1)-COORD(I,1))/ND                                     QMG01650
      DY=(COORD(J,2)-COORD(I,2))/ND                                     QMGC1660
      DZ=(COORD(J,3)-COORD(I,3))/ND                                     QMG01670
      DO 70 K=1,ND                                                      QMGC1680
      L=I+K                                                             QMG01690
      KL=KODE(L)                                                        QMG01700
      IF(KL.GT.0) GO TO 70                                              QMG01710
      KODE(L)=1                                                         QMG01720
```

```
      COORD(L,1)=COORD(I,1)+K*DX                                        QMG01730
      COORD(L,2)=COORD(I,2)+K*DY                                        QMG01740
      COORD(L,3)=COORD(I,3)+K*DZ                                        QMG01750
   70 CONTINUE                                                          QMG01760
   80 IF(NDX.EQ.0) GO TO 120                                            QMG01770
      DO 110 LA=1,NDX                                                   QMG01780
      READ(5,1200) I,J,N1,M1,L1                                         QMG01790
      MD=2*M1                                                           QMG01800
      DX=(COORD(J,1)-COORD(I,1))/MD                                     QMG01810
      DY=(COORD(J,2)-COORD(I,2))/MD                                     QMG01820
      DZ=(COORD(J,3)-COORD(I,3))/MD                                     QMG01830
      DO 110 K=1,M1                                                     QMG01840
      K1=K-1                                                            QMG01850
      K2=2*K                                                            QMG01860
      K3=K2-1                                                           QMG01870
      L=I+K*NY5                                                         QMG01880
      L2=I+NY4+K1*NY5-N1                                                QMG01890
      KL=KODE(L)                                                        QMG01900
      KL2=KODE(L2)                                                      QMG01910
      IF(KL2.GT.0) GO TO 90                                             QMG01920
      KODE(L2)=1                                                        QMG01930
      COORD(L2,1)=COORD(I,1)+K3*DX                                      QMG01940
      COORD(L2,2)=COORD(I,2)+K3*DY                                      QMG01950
      COORD(L2,3)=COORD(I,3)+K3*DZ                                      QMG01960
   90 IF(KL.GT.0) GO TO 110                                             QMG01970
      KODE(L)=1                                                         QMG01980
      COORD(L,1)=COORD(I,1)+K2*DX                                       QMG01990
      COORD(L,2)=COORD(I,2)+K2*DY                                       QMG02000
      COORD(L,3)=COORD(I,3)+K2*DZ                                       QMG02010
  110 CONTINUE                                                          QMG02020
  120 IF(NDZ.EQ.0) GO TO 150                                            QMG02030
      DO 140 LA=1,NDZ                                                   QMG02040
      READ(5,1200) I,J,N1,M1,L1                                         QMG02050
      LD=2*L1                                                           QMG02060
      DX=(COORD(J,1)-COORD(I,1))/LD                                     QMG02070
      DY=(COORD(J,2)-COORD(I,2))/LD                                     QMG02080
      DZ=(COORD(J,3)-COORD(I,3))/LD                                     QMG02090
      DO 140 K=1,L1                                                     QMG02100
      K1=K-1                                                            QMG02110
      K2=2*K                                                            QMG02120
      K3=K2-1                                                           QMG02130
      L=I+K*NPL                                                         QMG02140
      L2=I+NX*NY5+NY4*(1-M1)-N1+K1*NPL                                  QMG02150
      KL=KODE(L)                                                        QMG02160
      KL2=KODE(L2)                                                      QMG02170
      IF(KL2.GT.0) GO TO 130                                            QMG02180
      KODE(L2)=1                                                        QMG02190
      COORD(L2,1)=COORD(I,1)+K3*DX                                      QMG02200
```

```
      COORD(L2,2)=COORD(I,2)+K3*DY                                      QMG02210
      COORD(L2,3)=COORD(I,3)+K3*DZ                                      QMG02220
  130 IF(KL.GT.0) GO TO 140                                             QMG02230
      KODE(L)=1                                                         QMG02240
      COORD(L,1)=COORD(I,1)+K2*DX                                       QMG02250
      COORD(L,2)=COORD(I,2)+K2*DY                                       QMG02260
      COORD(L,3)=COORD(I,3)+K2*DZ                                       QMG02270
  140 CONTINUE                                                          QMG02280
C
C     COMPUTE COORDINATES OF CORNER NODES
C
C     FRONT AND BACK FACES
C
  150 IF((NY.EQ.1).OR.(NX.EQ.1)) GO TO 165                              QMG02285
      DO 160 L1=1,2                                                     QMG02290
      L2=L1-1                                                           QMG02300
      DO 160 ITER=1,60                                                  QMG02310
      DO 160 I=1,NX1                                                    QMG02320
      IC=(I-1)*NY                                                       QMG02330
      DO 160 J=1,NY6                                                    QMG02340
      IL1=IC+J+(NEL-NX*NY)*L2                                           QMG02350
      IL2=IL1+1                                                         QMG02360
      IL3=IL1+NY                                                        QMG02370
      I1=1+12*L2                                                        QMG02380
      I2=I1+4                                                           QMG02390
      I3=I1+6                                                           QMG02400
      JT1=NCON(IL1,I3)                                                  QMG02410
      IF(KODE(JT1).GT.0) GO TO 160                                      QMG02420
      JTA=NCON(IL1,I1)                                                  QMG02430
      JTB=NCON(IL1,I2)                                                  QMG02440
      JTC=NCON(IL2,I3)                                                  QMG02450
      JTD=NCON(IL3,I3)                                                  QMG02460
      DO 160 N=1,3                                                      QMG02470
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP       QMG02480
     1-COORD(JT1,N)                                                     QMG02490
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX                                  QMG02500
  160 CONTINUE                                                          QMG02510
C
C     LEFT AND RIGHT FACES
C
  165 IF((NY.EQ.1).OR.(NZ.EQ.1)) GO TO 175                              QMG02515
      DO 170 L1=1,2                                                     QMG02520
      L2=2*(L1-1)                                                       QMG02530
      L3=L1-1                                                           QMG02540
      DO 170 ITER=1,60                                                  QMG02550
      DO 170 I=1,NZ1                                                    QMG02560
      IM=I-1                                                            QMG02570
      DO 170 J=1,NY6                                                    QMG02580
```

```
      IL1=J+NY*(NX*(IM+L3)-L3)                                          QMG02590
      IL2=IL1+1                                                         QMG02600
      IL3=IL1+NX*NY                                                     QMG02610
      I1=15-L2                                                          QMG02620
      I2=5+L2                                                           QMG02630
      I3=17+L2                                                          QMG02640
      JT1=NCON(IL1,I3)                                                  QMG02650
      IF(KODE(JT1).GT.0) GO TO 170                                      QMG02660
      JTA=NCON(IL1,I1)                                                  QMG02670
      JTB=NCON(IL1,I2)                                                  QMG02680
      JTC=NCON(IL2,I3)                                                  QMG02690
      JTD=NCON(IL3,I3)                                                  QMG02700
      DO 170 N=1,3                                                      QMG02710
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP       QMG02720
     1-COORD(JT1,N)                                                     QMG02730
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX                                  QMG02740
  170 CONTINUE                                                          QMG02750
C
C     TOP AND BOTTOM FACES
C
  175 IF((NX.EQ.1).OR.(NZ.EQ.1)) GO TO 185                              QMG02755
      DO 180 L1=1,2                                                     QMG02760
      L2=L1-1                                                           QMG02770
      DO 180 ITER=1,60                                                  QMG02780
      DO 180 I=1,NZ1                                                    QMG02790
      IC=(I-1)*NX*NY+L2*NY6+1                                           QMG02800
      DO 180 J=1,NX1                                                    QMG02810
      IL1=IC+NY*(J-1)                                                   QMG02820
      IL2=IL1+NX*NY                                                     QMG02830
      IL3=IL1+NY                                                        QMG02840
      I1=1+6*L2                                                         QMG02850
      I2=15+2*L2                                                        QMG02860
      I3=I1+12                                                          QMG02870
      JT1=NCON(IL1,I3)                                                  QMG02880
      IF(KODE(JT1).GT.0) GO TO 180                                      QMG02890
      JTA=NCON(IL1,I2)                                                  QMG02900
      JTB=NCON(IL1,I1)                                                  QMG02910
      JTC=NCON(IL2,I3)                                                  QMG02920
      JTD=NCON(IL3,I3)                                                  QMG02930
      DO 180 N=1,3                                                      QMG02940
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP       QMG02950
     1-COORD(JT1,N)                                                     QMG02960
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX                                  QMG02970
  180 CONTINUE                                                          QMG02980
C
C     INTERIOR FACES
C
  185 IF((NY.EQ.1).OR.(NX.EQ.1)) GO TO 195                              QMG02985
```

```
      DO 190 I=1,NZ1                                                    QMG02990
      DO 190 ITER=1,60                                                  QMG03000
      DO 190 J=1,NX1                                                    QMG03010
      IC=(J-1)*NY+(I-1)*NX*NY                                           QMG03020
      DO 190 K=1,NY6                                                    QMG03030
      IL1=IC+K                                                          QMG03040
      IL2=IL1+1                                                         QMG03050
      IL3=IL1+NY                                                        QMG03060
      JT1=NCON(IL1,19)                                                  QMG03070
      IF(KODE(JT1).GT.0) GO TO 190                                      QMG03080
      JTA=NCON(IL1,13)                                                  QMG03090
      JTB=NCON(IL1,17)                                                  QMG03100
      JTC=NCON(IL2,19)                                                  QMG03110
      JTD=NCON(IL3,19)                                                  QMG03120
      DO 190 N=1,3                                                      QMG03130
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP       QMG03140
     1-COORD(JT1,N)                                                     QMG03150
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX                                  QMG03160
  190 CONTINUE                                                          QMG03170
C
C     COMPUTE COORDINATES OF MID-SIDE NODES
C
  195 DO 220 L=1,NEL                                                    QMG03180
      K=0                                                               QMG03190
      DO 200 I=1,7,2                                                    QMG03200
      I1=I+12                                                           QMG03210
      I2=I1-(4+K)                                                       QMG03220
      K=K+1                                                             QMG03230
      JT1=NCON(L,I2)                                                    QMG03240
      IF(KODE(JT1).GT.0) GO TO 200                                      QMG03250
      JT2=NCON(L,I)                                                     QMG03260
      JT3=NCON(L,I1)                                                    QMG03270
      DO 200 N=1,3                                                      QMG03280
      COORD(JT1,N)=COORD(JT2,N)-(COORD(JT2,N)-COORD(JT3,N))/2.0D0       QMG03290
  200 CONTINUE                                                          QMG03300
      DO 210 I=1,2                                                      QMG03310
      DO 210 J=1,5,4                                                    QMG03320
      I1=J+(I-1)*12                                                     QMG03330
      I2=I1+1                                                           QMG03340
      I3=I2+1                                                           QMG03350
      JT1=NCON(L,I2)                                                    QMG03360
      IF(KODE(JT1).GT.0) GO TO 210                                      QMG03370
      JT2=NCON(L,I1)                                                    QMG03380
      JT3=NCON(L,I3)                                                    QMG03390
      DO 210 N=1,3                                                      QMG03400
      COORD(JT1,N)=COORD(JT2,N)-(COORD(JT2,N)-COORD(JT3,N))/2.0D0       QMG03410
  210 CONTINUE                                                          QMG03420
      DO 220 I=1,2                                                      QMG03430
```

```
      K=6                                                               QMG03440
      K1=1                                                              QMG03450
      DO 220 J=1,3,2                                                    QMG03460
      I1=J+(I-1)*12                                                     QMG03470
      I2=I1+K                                                           QMG03480
      I3=I2+K1                                                          QMG03490
      K=2                                                               QMG03500
      K1=-1                                                             QMG03510
      JT1=NCON(L,I3)                                                    QMG03520
      IF(KODE(JT1).GT.0) GO TO 220                                      QMG03530
      JT2=NCON(L,I1)                                                    QMG03540
      JT3=NCON(L,I2)                                                    QMG03550
      DO 220 N=1,3                                                      QMG03560
      COORD(JT1,N)=COORD(JT2,N)-(COORD(JT2,N)-COORD(JT3,N))/2.0D0       QMG03570
  220 CONTINUE                                                          QMG03580
      WRITE(6,2000)                                                     QMG03590
      DO 240 L=1,NUMNP                                                  QMG03600
      KODE(L) = 0                                                       QMG03605
      IF(KORD.EQ.0) GO TO 230                                           QMG03610
      PHI=PI*COORD(L,2)                                                 QMG03620
      RAD=COORD(L,1)                                                    QMG03630
      COORD(L,1)=RAD*DCOS(PHI)                                          QMG03640
      COORD(L,2)=RAD*DSIN(PHI)                                          QMG03650
  230 IF(NCARD.EQ.0) GO TO 240                                          QMG03660
      WRITE(7,1800) L,(COORD(L,LP),LP=1,3)                              QMG03670
  240 WRITE(6,1800) L,(COORD(L,LP),LP=1,3)                              QMG03680
      READ(5,1900) MAP,NEFP,SGZ,UDP                                     QMG03690
      IF((SGZ.EQ.0.0).AND.(NEFP.EQ.0)) GO TO 330                        QMG03700
      IF(SGZ.EQ.0.0) GO TO 280                                          QMG03710
C
C     DETERMINE CONSISTENT GRAVITY LOAD
C
      DO 260 I=1,NEL                                                    QMG03720
      DO 250 L=1,NPEL                                                   QMG03730
      J2=NCON(I,L)                                                      QMG03740
      DO 250 K=1,NDF                                                    QMG03750
  250 COREL(L,K) = COORD(J2,K)                                          QMG03760
      CALL CUB4V                                                        QMG03790
      DO 260 J=1,NPEL                                                   QMG03800
      J1=NCON(I,J)                                                      QMG03810
      KODE(J1) = 1                                                      QMG03820
      GLV(J1)=GLV(J1)+SGZ*AK1(J)                                        QMG03830
  260 CONTINUE                                                          QMG03840
      WRITE(6,2100)                                                     QMG03850
      DO 270 I=1,NUMNP                                                  QMG03860
      IF((NCARD.EQ.0).OR.(NEFP.NE.0)) GO TO 270                         QMG03870
      WRITE(7,2400) I,GLV(I)                                            QMG03880
  270 WRITE(6,2300) I,GLV(I)                                            QMG03890
```

```
C                                                            QMG03900
C     DETERMINE CONSISTENT PRESSURE LOAD                     QMG03910
C                                                            QMG03920
  280 IF(NEFP.EQ.0) GO TO 330                                QMG03930
      WRITE(6,2200) UDP                                      QMG03940
      DO 300 I=1,NEFP                                        QMG03950
      READ(5,2700) NELP,NFACE                                QMG03960
      WRITE(6,2900) NELP,NFACE                               QMG03970
      DO 290 J=1,8                                           QMG03971
      J1=NC(NFACE,J)                                         QMG03972
  290 NELC(J)=NCON(NELP,J1)                                  QMG03973
      DO 295 L=1,NPEL                                        QMG03974
      J2 = NCON(NELP,L)                                      QMG03980
      DO 295 K=1,NDF                                         QMG03990
  295 COREL(L,K) = COORD(J2,K)                               QMG04000
      CALL QUAD5                                             QMG04010
      DO 300 J=1,8                                           QMG04020
      J1=NELC(J)                                             QMG04030
      KODE(J1) = 1                                           QMG04040
      DO 300 L=1,NDF                                         QMG04050
  300 PLM(J1,L)=PLM(J1,L)+UDP*AK3(J,L)                       QMG04060
      WRITE(6,2500)                                          QMG04070
      DO 320 I=1,NUMNP                                       QMG04080
      IF(KODE(I).EQ.0) GO TO 320                             QMG04090
      IF(NCARD.EQ.0) GO TO 310                               QMG04100
      PLM(I,3)=PLM(I,3)+GLV(I)                               QMG04110
      WRITE(7,2600) I,(PLM(I,J),J=1,3)                       QMG04120
  310 WRITE(6,1800) I,(PLM(I,L),L=1,3)                       QMG04130
  320 CONTINUE
  330 CONTINUE
      WRITE(6,1000)
C                                                            QMG04140
C     DRAW STRUCTURE MESH                                    QMG04150
C                                                            QMG04160
      IF(MAP.EQ.0) GO TO 340                                 QMG04170
      CALL GRID(NX,NY,NZ,NUMNP,NPL)                          QMG04180
  340 CONTINUE
      GO TO 10
      END
```

```
      SUBROUTINE CUB4V                                                  CUB00010
      IMPLICIT REAL*8(A-H,O-Z)                                          CUB00020
      CCMMON /NB1/ NPEL,NEL,NFACE,NUMNP                                 CUB00030
      COMMON /B3/ AK(20),AK1(20)                                        CUB00040
      DIMENSION AI(4),XI(4),AIA(4,4,4)                                  CUB00050
      DATA XI/0.8611363115940526, 0.3399810435848563,                   CUB00060
     1       -0.3399810435848563,-0.8611363115940526/                   CUB00070
      DATA AI/0.3478548451374539,0.6521451548625461,                    CUB00080
     1        0.6521451548625461,0.3478548451374539/                    CUB00090
      DO 100 I=1,NPEL                                                   CUB00100
  100 AK1(I)=0.0D0                                                      CUB00110
      DO 200 I=1,4                                                      CUB00120
      DO 200 J=1,4                                                      CUB00130
      DO 200 K=1,4                                                      CUB00140
  200 AIA(I,J,K)=AI(I)*AI(J)*AI(K)                                      CUB00150
      DO 400 I=1,4                                                      CUB00160
      X=XI(I)                                                           CUB00170
      DO 400 J=1,4                                                      CUB00180
      Y=XI(J)                                                           CUB00190
      DO 400 K=1,4                                                      CUB00200
      Z=XI(K)                                                           CUB00210
      CALL GRAP(X,Y,Z,1)                                                CUB00220
      DO 300 L=1,NPEL                                                   CUB00230
  300 AK1(L)=AK1(L)+AIA(I,J,K)*AK(L)                                    CUB00240
  400 CONTINUE                                                          CUB00250
      RETURN                                                            CUB00260
      END                                                               CUB00270
```

```
      SUBROUTINE QUAD5                                                  QAD00010
      IMPLICIT REAL*8(A-H,O-Z)                                          QAD00020
      DIMENSION AIA(5,5),AI(5),XI(5)                                    QAD00030
      COMMON /NB1/ NPEL,NEL,NFACE,NUMNP                                 QAD00040
      COMMON /B4/ AK2(8,3),AK3(8,3)                                     QAD00050
      DATA XI/0.906179845938664D,0.5384693101056831,0.0D0,-0.538469310100QAD00060
     156831,-0.906179845938664D/                                        QAD00070
      DATA AI/0.2369268850561891,0.4786286704993665,0.568888888888889,0QAD00080
     1.4786286704993665,0.2369268850561891/                             QAD00090
      DO 100 I=1,8                                                      QAD00100
      DO 100 J=1,3                                                      QAD00110
  100 AK3(I,J)=0.0D0                                                    QAD00120
      DO 200 I=1,5                                                      QAD00130
      DO 200 J=1,5                                                      QAD00140
  200 AIA(I,J)=AI(I)*AI(J)                                              QAD00150
      A=1.0D0                                                           QAD00160
      IF(NFACE.GT.3) A=-1.0D0                                           QAD00170
      DO 400 I=1,5                                                      QAD00180
      B=XI(I)                                                           QAD00190
      DO 400 J=1,5                                                      QAD00200
      C=XI(J)                                                           QAD00210
      GO TO (210,220,230,210,220,230),NFACE                             QAD00220
  210 X=A                                                               QAD00230
      Y=B                                                               QAD00240
      Z=C                                                               QAD00250
      GO TO 240                                                         QAD00260
  220 X=C                                                               QAD00270
      Y=A                                                               QAD00280
      Z=B                                                               QAD00290
      GO TO 240                                                         QAD00300
  230 X=B                                                               QAD00310
      Y=C                                                               QAD00320
      Z=A                                                               QAD00330
  240 CONTINUE                                                          QAD00340
      CALL GRAP(X,Y,Z,2)                                                QAD00350
      DO 300 K=1,8                                                      QAD00360
      DO 300 L=1,3                                                      QAD00370
  300 AK3(K,L)=AK3(K,L)+AIA(I,J)*AK2(K,L)                               QAD00380
  400 CONTINUE                                                          QAD00390
      RETURN                                                            QAD00400
      END                                                               QAD00410
```

```
      SUBROUTINE GRAP(X,Y,Z,N)                                          GRP00010
      IMPLICIT REAL*8(A-H,O-Z)                                          GRP00020
      COMMON /NB1/ NPEL,NEL,NFACE,NUMNP                                 GRP00030
      COMMON /B2/ COORD(999,3),COREL(20,3)                              GRP00040
      COMMON /B3/ AK(20),AK1(20)                                        GRP00050
      COMMON /B4/ AK2(8,3),AK3(8,3)                                     GRP00060
      DIMENSION W1(3,20),W2(20),W3(8),AJ(3,3),CORDG(20,3),              GRP00070
     1CRP(3),NC(6,8)                                                    GRP00080
      DATA CORDG/1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,1.0D0,1.0D0,    GRP00090
     11.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0, GRP00100
     21.0D0,1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,0.0D0,   GRP00110
     31.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,1.0D0,0.0D0,-1.0D0,-1.0D0,  GRP00120
     4-1.0D0,0.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,     GRP00130
     50.0D0,0.0D0,0.0D0,0.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,GRP00140
     6-1.0D0,-1.0D0/                                                    GRP00150
      DATA NC/13,13,1,3,17,15,9,14,2,10,18,14,1,15,3,15,19,13,8,10,4,16,GRP00160
     112,20,7,3,5,17,7,19,12,2,6,11,6,18,19,1,7,5,5,17,20,9,8,4,11,16/  GRP00170
      DF(C,D,E,X1,Y1,Z1)=X1*(1.0D0+D*Y1)*(1.0D0+E*Z1)*(2.0D0*C*X1+D*Y1+ GRP00180
     1E*Z1-1.0D0)/8.0D0                                                 GRP00190
      D2(C,D,E,    Y1,Z1)=-C*(1.0D0+D*Y1)*(1.0D0+E*Z1)/2.0D0            GRP00200
      D4(C,  E,    Y1,Z1)=(1.0D0-C*C)*Y1*(1.0D0+E*Z1)/4.0D0             GRP00210
      SFC(A,B,C,X1,Y1,Z1)=(1.0D0+A*X1)*(1.0D0+B*Y1)*(1.0D0+C*Z1)*(A*X1+ GRP00220
     1B*Y1+C*Z1-2.0D0)/8.0D0                                            GRP00230
      SFM(A,B,C,    Y1,Z1)=(1.0D0-A*A)*(1.0D0+B*Y1)*(1.0D0+C*Z1)/4.0D0  GRP00240
      DO 10 I=1,NPEL                                                    GRP00250
   10 AK(I)=0.0D0                                                       GRP00260
      DO 20 K=1,2                                                       GRP00270
      II=12*(K-1)+1                                                     GRP00280
      IT=II+6                                                           GRP00290
      DO 20 L=II,IT,2                                                   GRP00300
      X1=CORDG(L,1)                                                     GRP00310
      Y1=CORDG(L,2)                                                     GRP00320
      Z1=CORDG(L,3)                                                     GRP00330
      W1(1,L)=DF(X,Y,Z,X1,Y1,Z1)                                        GRP00340
      W1(2,L)=DF(Y,Z,X,Y1,Z1,X1)                                        GRP00350
      W1(3,L)=DF(Z,X,Y,Z1,X1,Y1)                                        GRP00360
   20 W2(L)=SFC(X,Y,Z,X1,Y1,Z1)                                         GRP00370
      DO 30 K=1,2                                                       GRP00380
      II=(K-1)*12+2                                                     GRP00390
      IT=II+4                                                           GRP00400
      DO 30 L=II,IT,4                                                   GRP00410
      X1=CORDG(L,1)                                                     GRP00420
      Y1=CORDG(L,2)                                                     GRP00430
      Z1=CORDG(L,3)                                                     GRP00440
      W1(1,L)=D2(X,Y,Z,    Y1,Z1)                                       GRP00450
      W1(2,L)=D4(X,  Z,    Y1,Z1)                                       GRP00460
```

```
      W1(3,L)=D4(X,   Y,    Z1,Y1)                                  GRP00470
   30 W2(L)=SFM(X,Y,Z,    Y1,Z1)                                    GRP00480
      DO 40 K=1,2                                                   GRP00490
      II=(K-1)*12+4                                                 GRP00500
      IT=II+4                                                       GRP00510
      DO 40 L=II,IT,4                                               GRP00520
      X1=CORDG(L,1)                                                 GRP00530
      Y1=CORDG(L,2)                                                 GRP00540
      Z1=CORDG(L,3)                                                 GRP00550
      W1(1,L)=D4(Y,   Z,    X1,Z1)                                  GRP00560
      W1(2,L)=D2(Y,Z,X,     Z1,X1)                                  GRP00570
      W1(3,L)=D4(Y,   X,    Z1,X1)                                  GRP00580
   40 W2(L)=SFM(Y,Z,X,    Z1,X1)                                    GRP00590
      DO 50 L=9,12                                                  GRP00600
      X1=CORDG(L,1)                                                 GRP00610
      Y1=CORDG(L,2)                                                 GRP00620
      Z1=CORDG(L,3)                                                 GRP00630
      W1(1,L)=D4(Z,   Y,    X1,Y1)                                  GRP00640
      W1(2,L)=D4(Z,   X,    Y1,X1)                                  GRP00650
      W1(3,L)=D2(Z,X,Y,     X1,Y1)                                  GRP00660
   50 W2(L)=SFM(Z,X,Y,    X1,Y1)                                    GRP00670
      DO 60 I=1,3                                                   GRP00680
      DO 60 J=1,3                                                   GRP00690
      AJ(I,J)=0.0D0                                                 GRP00700
      DO 60 K=1,NPEL                                                GRP00710
   60 AJ(I,J)=AJ(I,J)+W1(I,K)*COREL(K,J)                            GRP00720
      DTJ=AJ(1,1)*AJ(2,2)*AJ(3,3)+AJ(1,2)*AJ(2,3)*AJ(3,1)+AJ(1,3)*AJ(2,1GRP00730
     1)*AJ(3,2)-AJ(3,1)*AJ(2,2)*AJ(1,3)-AJ(3,2)*AJ(2,3)*AJ(1,1)-AJ(3,3)*GRP00740
     2AJ(2,1)*AJ(1,2)                                               GRP00750
      IF(N.EQ.1) GO TO 130                                          GRP00760
      GO TO (70,80,90,70,80,90),NFACE                               GRP00770
   70 CRP(1)=AJ(2,2)*AJ(3,3)-AJ(2,3)*AJ(3,2)                        GRP00780
      CRP(2)=AJ(2,3)*AJ(3,1)-AJ(2,1)*AJ(3,3)                        GRP00790
      CRP(3)=AJ(2,1)*AJ(3,2)-AJ(2,2)*AJ(3,1)                        GRP00800
      GO TO 100                                                     GRP00810
   80 CRP(1)=AJ(1,3)*AJ(3,2)-AJ(1,2)*AJ(3,3)                        GRP00820
      CRP(2)=AJ(1,1)*AJ(3,3)-AJ(1,3)*AJ(3,1)                        GRP00830
      CRP(3)=AJ(1,2)*AJ(3,1)-AJ(1,1)*AJ(3,2)                        GRP00840
      GO TO 100                                                     GRP00850
   90 CRP(1)=AJ(1,2)*AJ(2,3)-AJ(1,3)*AJ(2,2)                        GRP00860
      CRP(2)=AJ(1,3)*AJ(2,1)-AJ(1,1)*AJ(2,3)                        GRP00870
      CRP(3)=AJ(1,1)*AJ(2,2)-AJ(1,2)*AJ(2,1)                        GRP00880
  100 CONTINUE                                                      GRP00890
      SIGN=+1.0D0                                                   GRP00900
      IF(NFACE.GT.3) SIGN=-1.0D0                                    GRP00910
      DO 120 I=1,8                                                  GRP00920
      I1=NC(NFACE,I)                                                GRP00930
      W3(I)=W2(I1)                                                  GRP00940
```

```
      DO 120 J=1,3                                              GRP00950
  120 AK2(I,J)=CRP(J)*W3(I)*SIGN                                GRP00960
      GO TO 150                                                 GRP00970
  130 CONTINUE                                                  GRP00980
      DO 140 L=1,NPEL                                           GRP00990
  140 AK(L)=DTJ*W2(L)                                           GRP01000
  150 CONTINUE                                                  GRP01010
      RETURN                                                    GRP01020
      END                                                       GRP01030
```

```
      SUBROUTINE GRID(NX,NY,NZ,NUMNP,NPL)                              MAP00010
      IMPLICIT REAL*8(A-H,O-W)                                         MAP00020
      COMMON /B2/ COORD(999,3),COREL(20,3)                             MAP00030
      DIMENSION X(100),Y(100)                                          MAP00040
      REAL*8 LABEL/8H        /                                         MAP00050
      REAL*8 ITITLE(12)                                                MAP00060
 1000 FORMAT(6A8)                                                      MAP00070
      IWIDE=9                                                          MAP00080
      IHIGH=15                                                         MAP00090
      IXUP=15                                                          MAP00100
      NY1=NY+1                                                         MAP00110
      NX1=NX+1                                                         MAP00120
      NY2=2*NY+1                                                       MAP00130
      NX2=2*NX+1                                                       MAP00140
      NY5=3*NY+2                                                       MAP00150
      NZ1=NZ+1                                                         MAP00160
      VXMAX=-1.0D+10                                                   MAP00170
      VYMAX=-1.0D+10                                                   MAP00180
      VXMIN= 1.0D+10                                                   MAP00190
      VYMIN= 1.0D+10                                                   MAP00200
      DO 100 I=1,NUMNP                                                 MAP00210
      VXMAX=DMAX1(VXMAX,COORD(I,1))                                    MAP00220
      VYMAX=DMAX1(VYMAX,COORD(I,2))                                    MAP00230
      VXMIN=DMIN1(VXMIN,COORD(I,1))                                    MAP00240
  100 VYMIN=DMIN1(VYMIN,COORD(I,2))                                    MAP00250
      DO 200 I=1,NUMNP                                                 MAP00260
      COORD(I,1)=COORD(I,1)-VXMIN                                      MAP00270
  200 COORD(I,2)=COORD(I,2)-VYMIN                                      MAP00280
      XSCALE=((VXMAX-VXMIN)/9.0)*1E+10                                 MAP00290
      YSCALE=((VYMAX-VYMIN)/15.0)*1E+10                                MAP00300
      IF(XSCALE.LT.YSCALE) GO TO 300                                   MAP00310
      IHIGH=9                                                          MAP00320
      IXUP=9                                                           MAP00330
  300 IF(XSCALE.LE.YSCALE) XSCALE=YSCALE                               MAP00340
      IF(XSCALE.GT.YSCALE) YSCALE=XSCALE                               MAP00350
      IEXP=ALOG10(XSCALE)                                              MAP00360
      XSCALE=(XSCALE+(10**IEXP)*.5)*1E-10                              MAP00370
      DO 800 L1=1,NZ1                                                  MAP00380
      READ(5,1000) (ITITLE(I),I=1,6)                                   MAP00390
      READ(5,1000) (ITITLE(I),I=7,12)                                  MAP00400
      DO 600 I=1,NY1                                                   MAP00410
      I1=2*I-1+(L1-1)*NPL                                              MAP00420
      DO 500 J=1,NX2,2                                                 MAP00430
      J1=I1+NY5*(J-1)/2                                                MAP00440
      X(J)=COORD(J1,2)                                                 MAP00450
      Y(J)=-COORD(J1,1)                                                MAP00460
```

```
      IF(J.EQ.NX2) GO TO 500                                            MAP00470
      J2=J1+NY2-I+1                                                     MAP00480
      J3=J+1                                                            MAP00490
      X(J3)=COORD(J2,2)                                                 MAP00500
      Y(J3)=-COORD(J2,1)                                                MAP00510
  500 CONTINUE                                                          MAP00520
      MODCUR=2                                                          MAP00530
      IF(I.EQ.1) MODCUR=1                                               MAP00540
  600 CALL DRAW(NX2, X, Y,MODCUR,0,LABEL,ITITLE,XSCALE,XSCALE,IXUP,     MAP00550
     10,2,2,IWIDE,IHIGH,0,LAST)                                         MAP00560
      DO 800 I=1,NX1                                                    MAP00570
      I1=(I-1)*NY5+(L1-1)*NPL                                           MAP00580
      DO 700 J=1,NY2,2                                                  MAP00590
      J1=I1+J                                                           MAP00600
      X(J)=COORD(J1,2)                                                  MAP00610
      Y(J)=-COORD(J1,1)                                                 MAP00620
      IF(J.EQ.NY2) GO TO 700                                            MAP00630
      J2=J1+1                                                           MAP00640
      J3=J+1                                                            MAP00650
      X(J3)=COORD(J2,2)                                                 MAP00660
      Y(J3)=-COORD(J2,1)                                                MAP00670
  700 CONTINUE                                                          MAP00680
      IF(I.EQ.NX1) MODCUR=3                                             MAP00690
  800 CALL DRAW(NY2, X, Y,MODCUR,0,LABEL,ITITLE,XSCALE,XSCALE,IXUP,     MAP00700
     10,2,2,IWIDE,IHIGH,0,LAST)                                         MAP00710
      RETURN                                                            MAP00720
      END                                                               MAP00730
```

APPENDIX D
COMPUTER LISTING (CUMEG)

```
C
C     ****************************************************************
C     *                                                              *
C     *   MESH GENERATING PROGRAM FOR TRISOP                         *
C     *   USING CUBIC ELEMENTS                                       *
C     *   CODED BY D.E. HANSON, MAY 1971                             *
C     *   FRONT FACE IS IN XY PLANE, LEFT FACE IS IN YZ PLANE        *
C     *   NODAL POINT ONE IS AT UPPER LEFT CORNER OF FRONT FACE      *
C     *                                                              *
C     ****************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)                                          CMG00030
      DIMENSION GLV(1050),PLM(1050,3),NELC(12)                          CMG00040
      DIMENSION NCON(200,33),KODE(1050),TITLE(10),NC(6,12)              CMG00050
      COMMON /NB1/ NPEL,NEL,NFACE,NUMNP                                 CMG00060
      COMMON /B2/ COORD(1050,3),COREL(32,3)                             CMG00070
      COMMON /B3/ AK(32),AK1(32)                                        CMG00080
      COMMON /B4/ AK2(12,3),AK3(12,3)                                   CMG00090
      DATA NC/21,24,1,4,27,27,17,18,2,14,28,26,13,14,3,18,29,25,1,4,4,24CMG00100
     1,30,24,12,3,5,25,20,23,11,2,6,26,16,22,10,1,7,27,10,21,16,13,8,19,CMG00110
     29,32,20,17,9,15,8,31,30,21,10,7,7,30,31,22,11,6,15,29,32,23,12,5, CMG00120
     319,28/                                                            CMG00130
 1000 FORMAT('1')                                                       CMG00140
 1100 FORMAT (7I10,2I5)                                                 CMG00150
 1200 FORMAT (5I10)                                                     CMG00160
 1300 FORMAT(1I10,3F10.0)                                               CMG00170
 1400 FORMAT(5X,16I3,/,5X,16I3,1I5)                                     CMG00180
 1500 FORMAT(///,                                                       CMG00190
     115X,'NEL........TOTAL NUMBER OF ELEMENTS............'I5/,         CMG00200
     215X,'NUMNP......TOTAL NUMBER OF NODAL POINTS.........'I5/,        CMG00210
     315X,'NEQ........TOTAL NUMBER OF EQUATIONS............'I5/,        CMG00220
     415X,'NBAND......HALF-BAND WIDTH OF THE SYSTEM........'I5/,        CMG00230
     515X,'NN.........NUMBER OF BLOCKS PER COLUMN..........'I5/,        CMG00240
     615X,'MM.........NUMBER OF BLOCKS PER ROW.............'I5/,        CMG00250
     A15X,'TIME.......ESTIMATED SOLVE TIME (IN MINUTES)...'I5/,         CMG00260
     715X,'FT07F001...SPACE=(480,(',I5,',60))'/,                        CMG00270
     815X,'FT08F001...SPACE=(480,(',I5,',60))'/,                        CMG00280
     915X,'FT11F001...SPACE=(',I5,',(1,1))'/,                           CMG00290
     115X,'FT12F001...SPACE=(48,(',I5,',48))'/,                         CMG00300
    1215X,'FT13F001...SPACE=(48,(',I5,',48))'//)                        CMG00310
 1600 FORMAT(//,                                                        CMG00320
     125X,'CONNECTIVITY MATRIX'//,                                      CMG00330
     211X,'  EL   A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P',/,    CMG00340
     318X,'Q  R  S  T  U  V  W  X  Y  Z A1 B1 C1 D1 E1 F1   NMAT'//)    CMG00350
 1700 FORMAT(11X,I4,2X,16I3,/,17X,16I3,1I5/)                            CMG00360
 1800 FORMAT(6X,I10,3F15.5)                                             CMG00370
 1900 FORMAT(2I10,2F10.0)                                               CMG00380
 2000 FORMAT(///35X,'COORDINATES'//,14X,'NP', 9X,'X',14X,'Y',14X,'Z'/)  CMG00390
```

```
2100 FORMAT(///,20X,'GRAVITY LOAD',//,14X,'NP',11X,'Z'/)               CMG00400
2200 FORMAT(///22X,'PRESSURE INTENSITY =',F5.2,//27X,'EL',5X,'FACE'/)  CMG00410
2300 FORMAT(6X,I10,F20.12)                                              CMG00430
2400 FORMAT(I10,20X,F20.12)                                             CMG00440
2500 FORMAT(///,30X,'PRESSURE LOADS',//,14X,'NP', 9X,'X',14X,'Y',14X,  CMG00450
    1'Z',//)                                                            CMG00460
2600 FORMAT(I10,3F20.12)                                                CMG00470
2700 FORMAT(2I10)                                                       CMG00480
2800 FORMAT(10A8)                                                       CMG00490
2900 FORMAT(24X,I5,7X,I1)                                               CMG00500
  10 CONTINUE                                                           CMG00510
     READ(5,1100) NX,NY,NZ,NDX,NDY,NDZ,NPDP,KORD,NCARD                  CMG00520
     IF(NX.LT.0) STOP                                                   CMG00530
     READ(5,2800) TITLE                                                 CMG00540
     WRITE(6,2800) TITLE                                                CMG00550
     PI=3.141592653600/180.0D0                                          CMG00560
     ZRO=0.0D0                                                          CMG00570
     ORF=1.8D0                                                          CMG00580
     FP=4.0D0                                                           CMG00590
     NEL = NX*NY*NZ                                                     CMG00600
     NX1=NX-1                                                           CMG00610
     NY1=NY-1                                                           CMG00620
     NZ1=NZ-1                                                           CMG00630
     NY2=NY+1                                                           CMG00640
     NY3=3*NY+1                                                         CMG00650
     NY4=5*NY+3                                                         CMG00660
     NX2=NX+1                                                           CMG00670
     NPS=NX*NY4+NY3                                                     CMG00680
     NPM=2*NY2*NX2                                                      CMG00690
     NPL = NPS + NPM                                                    CMG00700
     NUMNP = NZ*NPL + NPS                                               CMG00710
C
C    COMPUTE ELEMENT CONNECTIVITY
C
     DO 20 IZ=1,NZ                                                      CMG00720
     IZM=IZ-1                                                           CMG00730
     DO 20 IX=1,NX                                                      CMG00740
     IXM=IX-1                                                           CMG00750
     IR=NX-IXM                                                          CMG00760
     IC=IXM*NY                                                          CMG00770
     I1=IXM*NY4                                                         CMG00780
     I2=I1+1                                                            CMG00790
     I3=I2+1                                                            CMG00800
     I4=I3+1                                                            CMG00810
     I5=I1+NY3                                                          CMG00820
     I6=I5+1                                                            CMG00830
     I7=I6+NY                                                           CMG00840
     I8=I7+1                                                            CMG00850
```

```
      I9=I8+NY                                                          CMG00860
      I10=I9+1                                                          CMG00870
      I11=I10+1                                                         CMG00880
      I12=I11+1                                                         CMG00890
      I13=I1+IR*NY4+NY3+IXM*NY2                                         CMG00900
      I14=I13+1                                                         CMG00910
      I15=I14+NY                                                        CMG00920
      I16=I15+1                                                         CMG00930
      I17=I13+NX2*NY2                                                   CMG00940
      I18=I17+1                                                         CMG00950
      I19=I18+NY                                                        CMG00960
      I20=I19+1                                                         CMG00970
      I21=I17+IXM*NY4+NY2*(IR+1)                                        CMG00980
      I22=I21+1                                                         CMG00990
      I23=I22+1                                                         CMG01000
      I24=I23+1                                                         CMG01010
      I25=I21+NY3                                                       CMG01020
      I26=I25+1                                                         CMG01030
      I27=I26+NY                                                        CMG01040
      I28=I27+1                                                         CMG01050
      I29=I28+NY                                                        CMG01060
      I30=I29+1                                                         CMG01070
      I31=I30+1                                                         CMG01080
      I32=I31+1                                                         CMG01090
      DO 20  IY=1,NY                                                    CMG01100
      IL=IZM*NX*NY+IC+IY                                                CMG01110
      IT=3*IY-2                                                         CMG01120
      NCON(IL, 1) = I9  + IT + IZM*NPL                                  CMG01130
      NCON(IL, 2) = I7  + IY + IZM*NPL                                  CMG01140
      NCON(IL, 3) = I5  + IY + IZM*NPL                                  CMG01150
      NCON(IL, 4) = I1  + IT + IZM*NPL                                  CMG01160
      NCON(IL, 5) = I2  + IT + IZM*NPL                                  CMG01170
      NCON(IL, 6) = I3  + IT + IZM*NPL                                  CMG01180
      NCON(IL, 7) = I4  + IT + IZM*NPL                                  CMG01190
      NCON(IL, 8) = I6  + IY + IZM*NPL                                  CMG01200
      NCON(IL, 9) = I8  + IY + IZM*NPL                                  CMG01210
      NCON(IL,10) = I12 + IT + IZM*NPL                                  CMG01220
      NCON(IL,11) = I11 + IT + IZM*NPL                                  CMG01230
      NCON(IL,12) = I10 + IT + IZM*NPL                                  CMG01240
      NCON(IL,13) = I15 + IY + IZM*NPL                                  CMG01250
      NCON(IL,14) = I13 + IY + IZM*NPL                                  CMG01260
      NCON(IL,15) = I14 + IY + IZM*NPL                                  CMG01270
      NCON(IL,16) = I16 + IY + IZM*NPL                                  CMG01280
      NCON(IL,17) = I19 + IY + IZM*NPL                                  CMG01290
      NCON(IL,18) = I17 + IY + IZM*NPL                                  CMG01300
      NCON(IL,19) = I18 + IY + IZM*NPL                                  CMG01310
      NCON(IL,20) = I20 + IY + IZM*NPL                                  CMG01320
      NCON(IL,21) = I29 + IT + IZM*NPL                                  CMG01330
```

```
      NCON(IL,22) = I27 + IY + IZM*NPL                                  CMG01340
      NCON(IL,23) = I25 + IY + IZM*NPL                                  CMG01350
      NCON(IL,24) = I21 + IT + IZM*NPL                                  CMG01360
      NCON(IL,25) = I22 + IT + IZM*NPL                                  CMG01370
      NCON(IL,26) = I23 + IT + IZM*NPL                                  CMG01380
      NCON(IL,27) = I24 + IT + IZM*NPL                                  CMG01390
      NCON(IL,28) = I26 + IY + IZM*NPL                                  CMG01400
      NCON(IL,29) = I28 + IY + IZM*NPL                                  CMG01410
      NCON(IL,30) = I32 + IT + IZM*NPL                                  CMG01420
      NCON(IL,31) = I31 + IT + IZM*NPL                                  CMG01430
      NCON(IL,32) = I30 + IT + IZM*NPL                                  CMG01440
      NCON(IL,33) = 1                                                   CMG01450
   20 CONTINUE                                                          CMG01460
C
C     COMPUTE BANDWIDTH, SOLUTION TIME, AND SPACE REQUIREMENTS
C                                                                       CMG01470
      NBAND=0                                                           CMG01480
      NS=60                                                             CMG01490
      SN=NS                                                             CMG01500
      NDF=3                                                             CMG01510
      NPEL=32                                                           CMG01520
      NST=NPEL*NDF                                                      CMG01530
      NEQ=NUMNP*NDF                                                     CMG01540
      NN=(NEQ+NS-1)/NS                                                  CMG01550
      NT=(NEQ+49)/50                                                    CMG01560
      DO 30 I=1,NEL                                                     CMG01570
      DO 30 J=1,31                                                      CMG01580
      JP=J+1                                                            CMG01590
      DO 30 K=JP,NPEL                                                   CMG01600
   30 NBAND=MAX0(NBAND,IABS(NCON(I,J)-NCON(I,K)))                       CMG01610
      NBAND=(NBAND+1)*NDF                                               CMG01620
      MM=(NBAND+2*(NS-1))/NS                                            CMG01630
      MT=(NBAND+98)/50                                                  CMG01640
      TK=(1.857D0*NT*MT**2+0.102D0*MT*NT**2-0.322D0*NT**2+4.682D0*NT*MT CMG01650
     1-5.23D0*MT**2)/60.0D0                                             CMG01660
      FT=(50.0D0/SN)**2                                                 CMG01670
      KT=TK*FT+0.5D0                                                    CMG01680
      K7=NN*(MM+1)*96                                                   CMG01690
      K8=NEL*96                                                         CMG01700
      K11=NEQ*8                                                         CMG01710
      K12=NEL*32                                                        CMG01720
      WRITE(6,1500) NEL,NUMNP,NEQ,NBAND,NN,MM,KT,K7,K8,K11,K12,K12      CMG01730
      WRITE(6,1600)                                                     CMG01740
      DO 50 I=1,NEL                                                     CMG01750
      IF(NCARD.EQ.0) GO TO 40                                           CMG01760
      WRITE(7,1400)(NCON(I,J), J = 1,33)                                CMG01770
   40 WRITE(6,1700) I,(NCON(I,J),J=1,33)                                CMG01780
   50 CONTINUE
```

```
      DO 60 I=1,NUMNP                                                   CMG01790
      GLV(I)=ZRO                                                        CMG01800
      KODE(I) =ZRO                                                      CMG01810
      DO 60 J=1,NDF                                                     CMG01820
      COORD(I,J)=ZRO                                                    CMG01830
   60 PLM(I,J)=ZRO                                                      CMG01840
      DO 70 J=1,NPDP                                                    CMG01850
      READ(5,1300) I,(COORD(I,IR),IR=1,3)                               CMG01860
   70 KODE(I)=1                                                         CMG01870
C
C     CCMPUTE COORDINATES OF EDGES
C
      IF(NDY.EQ.0) GO TO 90                                             CMG01880
      DO 80 LA=1,NDY                                                    CMG01890
      READ(5,1200) I,J,N1,M1,L1                                         CMG01900
      ND=3*N1                                                           CMG01910
      DX=(COORD(J,1)-COORD(I,1))/ND                                     CMG01920
      DY=(COORD(J,2)-COORD(I,2))/ND                                     CMG01930
      DZ=(COORD(J,3)-COORD(I,3))/ND                                     CMG01940
      DO 80 K=1,ND                                                      CMG01950
      L=I+K                                                             CMG01960
      IF(KODE(L).GT.0) GO TO 80                                         CMG01970
      KODE(L)=1                                                         CMG01980
      COORD(L,1)=COORD(I,1)+K*DX                                        CMG01990
      COORD(L,2)=COORD(I,2)+K*DY                                        CMG02000
      COORD(L,3)=COORD(I,3)+K*DZ                                        CMG02010
   80 CONTINUE                                                          CMG02020
   90 IF(NDX.EQ.0) GO TO 130                                            CMG02030
      DO 120 LA=1,NDX                                                   CMG02040
      READ(5,1200) I,J,N1,M1,L1                                         CMG02050
      MD=3*M1                                                           CMG02060
      DX=(COORD(J,1)-COORD(I,1))/MD                                     CMG02070
      DY=(COORD(J,2)-COORD(I,2))/MD                                     CMG02080
      DZ=(COORD(J,3)-COORD(I,3))/MD                                     CMG02090
      DO 120 K=1,M1                                                     CMG02100
      K1=K-1                                                            CMG02110
      K2=3*K-2                                                          CMG02120
      L=I+K*NY4                                                         CMG02130
      L3=L-(2*N1+1+NY)                                                  CMG02140
      L2=L3-NY2                                                         CMG02150
      IF(KODE(L2).GT.0) GO TO 100                                       CMG02160
      KODE(L2)=1                                                        CMG02170
      COORD(L2,1)=COORD(I,1)+K2*DX                                      CMG02180
      COORD(L2,2)=COORD(I,2)+K2*DY                                      CMG02190
      COORD(L2,3)=COORD(I,3)+K2*DZ                                      CMG02200
  100 IF(KODE(L3).GT.0) GO TO 110                                       CMG02210
      KODE(L3)=1                                                        CMG02220
      COORD(L3,1)=COORD(L2,1)+DX                                        CMG02230
```

```
                                                          CMG02240
      COORD(L3,2)=COORD(L2,2)+DY                          CMG02250
      COORD(L3,3)=COORD(L2,3)+DZ                          CMG02260
  110 IF(KODE(L).GT.0) GO TO 120                          CMG02270
      KODE(L)=1                                           CMG02280
      COORD(L,1)=COORD(L3,1)+DX                           CMG02290
      COORD(L,2)=COORD(L3,2)+DY                           CMG02300
      COORD(L,3)=COORD(L3,3)+DZ                           CMG02310
  120 CONTINUE                                            CMG02320
  130 IF(NDZ.EQ.0) GO TO 165                              CMG02340
      DO 160 LA=1,NDZ                                     CMG02350
      READ(5,1200) I,J,N1,M1,L1                           CMG02360
      LD=3*L1
      DX=(COORD(J,1)-COORD(I,1))/LD                       CMG02380
      DY=(COORD(J,2)-COORD(I,2))/LD                       CMG02390
      DZ=(COORD(J,3)-COORD(I,3))/LD                       CMG02400
      DO 160 K=1,L1                                       CMG02410
      K1=K-1                                              CMG02420
      K2=3*K-2                                            CMG02430
      L=I+K*NPL                                           CMG02440
      L3=L-(M1*NY4+2*N1+1+(NX-M1)*NY2+NY)                 CMG02450
      L2=L3-NX2*NY2                                       CMG02460
      IF(KODE(L2).GT.0) GO TO 140                         CMG02470
      KODE(L2)=1                                          CMG02480
      COORD(L2,1)=COORD(I,1)+K2*DX                        CMG02490
      COORD(L2,2)=COORD(I,2)+K2*DY                        CMG02500
      COORD(L2,3)=COORD(I,3)+K2*DZ                        CMG02510
  140 IF(KODE(L3).GT.0) GO TO 150                         CMG02520
      KODE(L3)=1                                          CMG02530
      COORD(L3,1)=COORD(L2,1)+DX                          CMG02540
      COORD(L3,2)=COORD(L2,2)+DY                          CMG02550
      COORD(L3,3)=COORD(L2,3)+DZ                          CMG02560
  150 IF(KODE(L).GT.0) GO TO 160                          CMG02570
      KODE(L)=1                                           CMG02580
      COORD(L,1)=COORD(L3,1)+DX                           CMG02590
      COORD(L,2)=COORD(L3,2)+DY                           CMG02600
      COORD(L,3)=COORD(L3,3)+DZ                           CMG02610
  160 CONTINUE
C
C     COMPUTE COORDINATES CF CORNER NODES
C
C     FRONT AND BACK FACES
C
  165 IF((NY.EQ.1).OR.(NX.EQ.1)) GO TO 175                CMG02620
      DO 170 L1=1,2                                       CMG02630
      L2=L1-1                                             CMG02640
      DO 170 ITER=1,60                                    CMG02650
      DO 170 I=1,NX1                                      CMG02660
      IC=(I-1)*NY                                         CMG02670
```

```
      DO 170  J=1,NY1                                                   CMG02680
      IL1=IC+(NEL-NX*NY)*L2                                             CMG02690
      IL2=IL1+1                                                         CMG02700
      IL3=IL1+NY                                                        CMG02710
      I1=1+20*L2                                                        CMG02720
      I2=I1+6                                                           CMG02730
      I3=I1+9                                                           CMG02740
      JT1=NCON(IL1,I3)                                                  CMG02750
      IF(KODE(JT1).GT.0) GO TO 170                                      CMG02760
      JTA=NCON(IL1,I1)                                                  CMG02770
      JTB=NCON(IL1,I2)                                                  CMG02780
      JTC=NCON(IL2,I3)                                                  CMG02790
      JTD=NCON(IL3,I3)                                                  CMG02800
      DO 170  N=1,3                                                     CMG02810
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP       CMG02820
     1-COORD(JT1,N)                                                     CMG02830
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX                                  CMG02840
  170 CONTINUE                                                          CMG02850
  175 IF((NY.EQ.1).OR.(NZ.EQ.1)) GO TO 185                              CMG02860
C
C     LEFT AND RIGHT FACES
C
      DO 180  L1=1,2                                                    CMG02870
      L2=3*(L1-1)                                                       CMG02880
      L3=L1-1                                                           CMG02890
      DO 180  ITER=1,60                                                 CMG02900
      DO 180  I=1,NZ1                                                   CMG02910
      IM=I-1                                                            CMG02920
      DO 180  J=1,NY1                                                   CMG02930
      IL1=J+NY*(NX*(IM+L3)-L3)                                          CMG02940
      IL2=IL1+1                                                         CMG02950
      IL3=IL1+NX*NY                                                     CMG02960
      I1=24-L2                                                          CMG02970
      I2=7+L2                                                           CMG02980
      I3=27+L2                                                          CMG02990
      JT1=NCON(IL1,I3)                                                  CMG03000
      IF(KODE(JT1).GT.0) GO TO 180                                      CMG03010
      JTA=NCON(IL1,I1)                                                  CMG03020
      JTB=NCON(IL1,I2)                                                  CMG03030
      JTC=NCON(IL2,I3)                                                  CMG03040
      JTD=NCON(IL3,I3)                                                  CMG03050
      DO 180  N=1,3                                                     CMG03060
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP       CMG03070
     1-COORD(JT1,N)                                                     CMG03080
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX                                  CMG03090
  180 CONTINUE                                                          CMG03100
  185 IF((NX.EQ.1).OR.(NZ.EQ.1)) GO TO 195                              CMG03110
C
```

```
C
C     TOP AND BOTTOM FACES
C
      DO 190 L1=1,2                                                      CMG03120
      L2=L1-1                                                            CMG03130
      DO 190 ITER=1,60                                                   CMG03140
      DO 190 I=1,NZ1                                                     CMG03150
      IC=(I-1)*NX*NY+L2*NY1+1                                            CMG03160
      DO 190 J=1,NX1                                                     CMG03170
      IL1=IC+NY*(J-1)                                                    CMG03180
      IL2=IL1+NX*NY                                                      CMG03190
      IL3=IL1+NY                                                         CMG03200
      I1=1+9*L2                                                          CMG03210
      I2=24+3*L2                                                         CMG03220
      I3=I1+20                                                           CMG03230
      JT1=NCON(IL1,I3)                                                   CMG03240
      IF(KODE(JT1).GT.0) GO TO 190                                       CMG03250
      JTA=NCON(IL1,I2)                                                   CMG03260
      JTB=NCON(IL1,I1)                                                   CMG03270
      JTC=NCON(IL2,I3)                                                   CMG03280
      JTD=NCON(IL3,I3)                                                   CMG03290
      DO 190 N=1,3                                                       CMG03300
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP        CMG03310
     1-COORD(JT1,N)                                                      CMG03320
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX                                   CMG03330
  190 CONTINUE                                                           CMG03340
  195 IF((NY.EQ.1).OR.(NX.EQ.1)) GO TO 205                               CMG03350
C
C     INTERIOR FACES
C
      DO 200 I=1,NZ1                                                     CMG03360
      DO 200 ITER=1,60                                                   CMG03370
      DO 200 J=1,NX1                                                     CMG03380
      IC=(J-1)*NY+(I-1)*NX*NY                                            CMG03390
      DO 200 K=1,NY1                                                     CMG03400
      IL1=IC+K                                                           CMG03410
      IL2=IL1+1                                                          CMG34320
      IL3=IL1+NY                                                         CMG03430
      JT1=NCON(IL1,30)                                                   CMG03440
      IF(KODE(JT1).GT.0) GO TO 200                                       CMG03450
      JTA=NCON(IL1,21)                                                   CMG03460
      JTB=NCON(IL1,27)                                                   CMG03470
      JTC=NCON(IL2,30)                                                   CMG03480
      JTD=NCON(IL3,30)                                                   CMG03490
      DO 200 N=1,3                                                       CMG03500
      DX=(COORD(JTA,N)+COORD(JTB,N)+COORD(JTC,N)+COORD(JTD,N))/FP        CMG03510
     1-COORD(JT1,N)                                                      CMG03520
      COORD(JT1,N)=COORD(JT1,N)+ORF*DX                                   CMG03530
  200 CONTINUE                                                           CMG03540
```

```
C
C     COMPUTE COORDINATES OF MID-SIDE NODES
C                                                                       CMG03550
  205 DO 230 L=1,NEL                                                    CMG03560
      K=12                                                              CMG03570
      DO 210 I=1,10,3                                                   CMG03580
      I1=I+20                                                           CMG03590
      I2=I+K                                                            CMG03600
      I3=I2+4                                                           CMG03610
      K=K-2                                                             CMG03620
      JT1=NCON(L,I2)                                                    CMG03630
      IF(KODE(JT1).GT.0) GO TO 210                                      CMG03640
      JT2=NCON(L,I3)                                                    CMG03650
      JT3=NCON(L,I1)                                                    CMG03660
      JT4=NCON(L,I)                                                     CMG03670
      DO 210 N=1,3                                                      CMG03680
      DX=(COORD(JT3,N)-COORD(JT4,N))/3.0D0                              CMG03690
      COORD(JT1,N)=COORD(JT4,N)+DX                                      CMG03700
        COORD(JT2,N)=COORD(JT1,N)+DX                                    CMG03710
  210 CONTINUE                                                          CMG03720
      DO 220 I=1,2                                                      CMG03730
      DO 220 J=1,7,6                                                    CMG03740
      I1=J+(I-1)*20                                                     CMG03750
      I2=I1+1                                                           CMG03760
      I3=I2+1                                                           CMG03770
      I4=I3+1                                                           CMG03780
      JT1=NCON(L,I2)                                                    CMG03790
      IF(KODE(JT1).GT.0) GO TO 220                                      CMG03800
      JT2=NCON(L,I3)                                                    CMG03810
      JT3=NCON(L,I4)                                                    CMG03820
      JT4=NCON(L,I1)                                                    CMG03830
      DO 220 N=1,3                                                      CMG03840
      DX=(COORD(JT3,N)-COORD(JT4,N))/3.0D0                              CMG03850
      COORD(JT1,N)=COORD(JT4,N)+DX                                      CMG03860
      COORD(JT2,N)=COORD(JT1,N)+DX                                      CMG03870
  220 CONTINUE                                                          CMG03880
      DO 230 I=1,2                                                      CMG03890
      K=9                                                               CMG03900
      K1=1                                                              CMG03910
      DO 230 J=1,4,3                                                    CMG03920
      I1=J+(I-1)*20                                                     CMG03930
      I2=I1+K                                                           CMG03940
      I3=I2+K1                                                          CMG03950
      I4=I3+K1                                                          CMG03960
      K=3                                                               CMG03970
      K1=-1                                                             CMG03980
      JT1=NCON(L,I3)                                                    CMG03990
      IF(KODE(JT1).GT.0) GO TO 230
```

```
      JT2=NCON(L,I4)                                                    CMG04000
      JT3=NCON(L,I1)                                                    CMG00410
      JT4=NCON(L,I2)                                                    CMG04020
      DO 230 N=1,3                                                      CMG00430
      DX=(COORD(JT3,N)-COORD(JT4,N))/3.0D0                              CMG04050
      COORD(JT1,N)=COORD(JT4,N)+DX                                      CMG04060
      COORD(JT2,N)=COORD(JT1,N)+DX                                      CMG04070
  230 CONTINUE                                                          CMG04080
      WRITE(6,2000)                                                     CMG04090
      DO 250 I=1,NUMNP                                                  CMG04100
      KODE(I) = 0                                                       CMG04105
      IF(KORD.EQ.0) GO TO 240                                           CMG04110
      PHI=PI*COORD(I,2)                                                 CMG04120
      RAD=COORD(I,1)                                                    CMG04130
      COORD(I,1)=RAD*DCOS(PHI)                                          CMG04140
      COORD(I,2)=RAD*DSIN(PHI)                                          CMG04150
  240 IF(NCARD.EQ.0) GO TO 250                                          CMG04160
      WRITE(7,1800) I,(COORD(I,L),L=1,3)                                CMG04170
  250 WRITE(6,1800) I,(COORD(I,L),L=1,3)                                CMG04180
      READ(5,1900) MAP,NEFP,SGZ,UDP                                     CMG04190
      IF((SGZ.EQ.0.0).AND.(NEFP.EQ.0)) GO TO 340                        CMG04200
      IF(SGZ.EQ.0.0) GO TO 290                                          CMG04210
C
C     DETERMINE CONSISTENT GRAVITY LOAD
C
      DO 270 I=1,NEL                                                    CMG04220
      DO 260 L=1,NPEL                                                   CMG04230
      J2 = NCON(I,L)                                                    CMG04240
      DO 260 K=1,NDF                                                    CMG04250
  260 COREL(L,K) = COORD(J2,K)                                          CMG04260
      CALL CUB5V                                                        CMG04290
      DO 270 J=1,NPEL                                                   CMG04300
      J1=NCON(I,J)                                                      CMG04310
      KODE(J1) = 1                                                      CMG04320
      GLV(J1)=GLV(J1)+SGZ*AK1(J)                                        CMG04330
  270 CONTINUE                                                          CMG04340
      WRITE(6,2100)                                                     CMG04350
      DO 280 I=1,NUMNP                                                  CMG04360
      IF((NCARD.EQ.0).OR.(NEFP.NE.0)) GO TO 280                         CMG04370
      WRITE(7,2400) I,GLV(I)                                            CMG04380
  280 WRITE(6,2300) I,GLV(I)                                            CMG04390
  290 IF(NEFP.EQ.0) GO TO 340                                           CMG04400
C
C     DETERMINE CONSISTENT PRESSURE LOAD
C
      WRITE(6,2200) UDP                                                 CMG04410
      DO 310 I=1,NEFP                                                   CMG04420
      READ(5,2700) NELP,NFACE                                           CMG04430
```

```
      WRITE(6,2900) NELP,NFACE                                          CMG04440
      DO 300 J=1,12                                                     CMG04450
      J1=NC(NFACE,J)                                                    CMG04460
  300 NELC(J)=NCON(NELP,J1)                                             CMG04470
      DO 305 L=1,NPEL                                                   CMG04471
      J2 = NCON(NELP,L)                                                 CMG04472
      DO 305 K=1,NDF                                                    CMG04473
  305 COREL(L,K) = COORD(J2,K)                                          CMG04474
      CALL QUAD5                                                        CMG04480
      DO 310 J=1,12                                                     CMG04490
      J1=NELC(J)                                                        CMG04500
      KODE(J1) = 1                                                      CMG04510
      DO 310 L=1,NDF                                                    CMG04520
  310 PLM(J1,L)=PLM(J1,L)+UDP*AK3(J,L)                                  CMG04530
      WRITE(6,2500)                                                     CMG04540
      DO 330 I=1,NUMNP                                                  CMG04550
      IF(KODE(I).EQ.0) GO TO 330                                        CMG04560
      IF(NCARD.EQ.0) GO TO 320                                          CMG04570
      PLM(I,3)=PLM(I,3)+GLV(I)                                          CMG04580
      WRITE(7,2600) I,(PLM(I,J),J=1,3)                                  CMG04590
  320 WRITE(6,1800) I,(PLM(I,L),L=1,3)                                  CMG04600
  330 CONTINUE                                                          CMG04610
  340 CONTINUE                                                          CMG04620
      WRITE(6,1000)                                                     CMG04630
      IF(MAP.EQ.0) GO TO 350                                            CMG04640
C
C     DRAW STRUCTURE MESH
C
      CALL GRID(NX,NY,NZ,NUMNP,NPL)                                     CMG04650
  350 CONTINUE                                                          CMG04660
      GO TO 10                                                          CMG04670
      END                                                               CMG04680
```

```
      SUBROUTINE CUB5V                                                  CUB00010
      IMPLICIT REAL*8(A-H,O-Z)                                          CUB00020
      COMMON /NB1/ NPEL,NEL,NFACE,NUMNP                                 CUB00030
      COMMON /B3/ AK(32),AK1(32)                                        CUB00040
      DIMENSION XI(5),AI(5),AIA(5,5,5)                                  CUB00050
      DATA XI/0.906179845938664D,0.5384693101056831,0.0D0,-0.53846931010CUB00060
     156831,-0.9061798459386640/                                        CUB00070
      DATA AI/0.2369268850561891,0.4786286704993665,0.5688888888888889,0CUB00080
     1.4786286704993665,0.2369268850561891/                             CUB00090
      DO 100 I=1,NPEL                                                   CUB00100
  100 AK1(I)=0.0D0                                                      CUB00110
      DO 200 I=1,5                                                      CUB00120
      DO 200 J=1,5                                                      CUB00130
      DO 200 K=1,5                                                      CUB00140
  200 AIA(I,J,K)=AI(I)*AI(J)*AI(K)                                      CUB00150
      DO 400 I=1,5                                                      CUB00160
      X=XI(I)                                                           CUB00170
      DO 400 J=1,5                                                      CUB00180
      Y=XI(J)                                                           CUB00190
      DO 400 K=1,5                                                      CUB00200
      Z=XI(K)                                                           CUB00210
      CALL GRAP(X,Y,Z,1)                                                CUB00220
      DO 300 L=1,NPEL                                                   CUB00230
  300 AK1(L)=AK1(L)+AIA(I,J,K)*AK(L)                                    CUB00240
  400 CONTINUE                                                          CUB00250
      RETURN                                                            CUB00260
      END                                                               CUB00270
```

```
      SUBROUTINE QUAD5
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AIA(5,5),AI(5),XI(5)
      COMMON /NB1/ NPEL,NEL,NFACE,NUMNP
      COMMON /B4/ AK2(12,3),AK3(12,3)
      DATA XI/0.9061798459386640,0.5384693101056831,0.0D0,-0.53846931010
     156831,-0.9061798459386640/
      DATA AI/0.2369268850561891,0.4786286704993665,0.5688888888888889,0
     1.4786286704993665,0.2369268850561891/
      DO 100 I=1,12
      DO 100 J=1,3
  100 AK3(I,J)=0.0D0
      DO 200 I=1,5
      DO 200 J=1,5
  200 AIA(I,J)=AI(I)*AI(J)
      A=1.0D0
      IF(NFACE.GT.3) A=-1.0D0
      DO 400 I=1,5
      B=XI(I)
      DO 400 J=1,5
      C=XI(J)
      GO TO (210,220,230,210,220,230),NFACE
  210 X=A
      Y=B
      Z=C
      GO TO 240
  220 X=C
      Y=A
      Z=B
      GO TO 240
  230 X=B
      Y=C
      Z=A
  240 CONTINUE
      CALL GRAP(X,Y,Z,2)
      DO 300 K=1,12
      DO 300 L=1,3
  300 AK3(K,L)=AK3(K,L)+AIA(I,J)*AK2(K,L)
  400 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRAP(X,Y,Z,N)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON /NB1/ NPEL,NEL,NFACE,NUMNP
      COMMON /B2/ COORD(1050,3),COREL(32,3)
      COMMON /B3/ AK(32),AK1(32)
      COMMON /B4/ AK2(12,3),AK3(12,3)
      DIMENSION W1(3,32),W2(32),W3(12),AJ(3,3),CORDG(32,3),
     1CRP(3),NC(6,12)
      DATA CORDG/1.0D0,.3333333333333333,-.3333333333333333,-1.0D0,
     1-1.0D0,-1.0D0,-1.0D0,-.3333333333333333,.3333333333333333,1.0D0,
     21.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,
     31.0D0,.3333333333333333,-.3333333333333333,-1.0D0,-1.0D0,-1.0D0,
     4-1.0D0,-.3333333333333333,.3333333333333333,1.0D0,1.0D0,1.0D0,
     51.0D0,1.0D0,1.0D0,1.0D0,.3333333333333333,-.3333333333333333,
     6-1.0D0,-1.0D0,-1.0D0,-1.0D0,-.3333333333333333,.3333333333333333,
     71.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,-1.0D0,-1.0D0,1.0D0,1.0D0,
     81.0D0,1.0D0,.3333333333333333,-.3333333333333333,-1.0D0,-1.0D0,
     9-1.0D0,-1.0D0,-.3333333333333333,.3333333333333333,1.0D0,1.0D0,
     A1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,1.0D0,
     B.3333333333333333,.3333333333333333,.3333333333333333,
     C .3333333333333333,-.3333333333333333,-.3333333333333333,
     D-.3333333333333333,-.3333333333333333,-1.0D0,-1.0D0,-1.0D0,-1.0D0,
     E-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0,-1.0D0/
      DATA NC/21,24,1,4,27,27,17,18,2,14,28,26,13,14,3,18,29,25,1,4,4,24
     1,30,24,12,3,5,25,20,23,11,2,6,26,16,22,10,1,7,27,10,21,16,13,8,19,
     29,32,20,17,9,15,8,31,30,21,10,7,7,30,31,22,11,6,15,29,32,23,12,5,
     319,28/
      DFX(X,Y,Z,X1,Y1,Z1)=(1.0D0+Y*Y1)*(1.0D0+Z*Z1)*(9.0D0*X1*(X*X+Y*Y+
     1Z*Z)-19.0D0*X1+18.0D0*X+18.0D0*X*X*X1)/64.0D0
      DX(X,Y,Z,X1,Y1,Z1)=9.0D0*(1.0D0+Y*Y1)*(1.0D0+Z*Z1)*(-2.0D0*X-27.0D
     10*X*X*X1+9.0D0*X1)/64.0D0
      DY(X,  Z,X1,Y1,Z1)=9.0D0*(1.0D0-X*X)*(1.0D0+9.0D0*X*X1)*Y1*(1.0D0+
     1Z*Z1)/64.0D0
      DZ(X,Y,  X1,Y1,Z1)=DY(X,  Y,X1,Z1,Y1)
      SFC(A,B,C,X1,Y1,Z1)=(1.0D0+A*X1)*(1.0D0+B*Y1)*(1.0D0+C*Z1)*(9.0D0*
     1(A*A+B*B+C*C)-19.0D0)/64.0D0
      SFM(A,B,C,X1,Y1,Z1)=9.0D0*(1.0D0-A*A)*(1.0D0+9.0D0*A*X1)*(1.0D0+
     1B*Y1)*(1.0D0+C*Z1)/64.0D0
      DO 10 I=1,NPEL
   10 AK(I)=0.0D0
       DO 20 I=1,2
      II=20*(I-1)+1
      IT=II+9
      DO 20 J=II,IT,3
      X1=CORDG(J,1)
      Y1=CORDG(J,2)
```

```
      Z1=CORDG(J,3)                                                     GRP00470
      W2(J)=SFC(X,Y,Z,X1,Y1,Z1)                                         GRP00480
      W1(1,J)=DFX(X,Y,Z,X1,Y1,Z1)                                       GRP00490
      W1(2,J)=DFX(Y,X,Z,Y1,X1,Z1)                                       GRP00500
   20 W1(3,J)=DFX(Z,X,Y,Z1,X1,Y1)                                       GRP00510
      DO 30 K=1,2                                                       GRP00520
      IK=20*(K-1)                                                       GRP00530
      DO 30 I=1,2                                                       GRP00540
      II=6*(I-1)+2+IK                                                   GRP00550
      IT=II+1                                                           GRP00560
      DO 30 J=II,IT                                                     GRP00570
      X1=CORDG(J,1)                                                     GRP00580
      Y1=CORDG(J,2)                                                     GRP00590
      Z1=CORDG(J,3)                                                     GRP00600
      W2(J)=SFM(X,Y,Z,X1,Y1,Z1)                                         GRP00610
      W1(1,J)=DX(X,Y,Z,X1,Y1,Z1)                                        GRP00620
      W1(2,J)=DY(X,  Z,X1,Y1,Z1)                                        GRP00630
   30 W1(3,J)=DZ(X,Y,  X1,Y1,Z1)                                        GRP00640
      DO 40 K=1,2                                                       GRP00650
      IK=20*(K-1)                                                       GRP00660
      DO 40 I=1,2                                                       GRP00670
      II=6*(I-1)+5+IK                                                   GRP00680
      IT=II+1                                                           GRP00690
      DO 40 J=II,IT                                                     GRP00700
      X1=CORDG(J,1)                                                     GRP00710
      Y1=CORDG(J,2)                                                     GRP00720
      Z1=CORDG(J,3)                                                     GRP00730
      W2(J)=SFM(Y,Z,X,Y1,Z1,X1)                                         GRP00740
      W1(1,J)=DY(Y,  Z,Y1,X1,Z1)                                        GRP00750
      W1(2,J)=DX(Y,X,Z,Y1,X1,Z1)                                        GRP00760
   40 W1(3,J)=DZ(Y,X,  Y1,X1,Z1)                                        GRP00770
      DO 50 K=1,2                                                       GRP00780
      IK=4*(K-1)                                                        GRP00790
      II=13+IK                                                          GRP00800
      IT=II+3                                                           GRP00810
      DO 50 J=II,IT                                                     GRP00820
      X1=CORDG(J,1)                                                     GRP00830
      Y1=CORDG(J,2)                                                     GRP00840
      Z1=CORDG(J,3)                                                     GRP00850
      W2(J)=SFM(Z,X,Y,Z1,X1,Y1)                                         GRP00860
      W1(1,J)=DZ(Z,Y,  Z1,Y1,X1)                                        GRP00870
      W1(2,J)=DY(Z,  X,Z1,Y1,X1)                                        GRP00880
   50 W1(3,J)=DX(Z,Y,X,Z1,Y1,X1)                                        GRP00890
      DO 60 I=1,3                                                       GRP00900
      DO 60 J=1,3                                                       GRP00910
      AJ(I,J)=0.0D0                                                     GRP00920
      DO 60 K=1,NPEL                                                    GRP00930
   60 AJ(I,J)=AJ(I,J)+W1(I,K)*COREL(K,J)                                GRP00940
```

```
      DTJ=AJ(1,1)*AJ(2,2)*AJ(3,3)+AJ(1,2)*AJ(2,3)*AJ(3,1)+AJ(1,3)*AJ(2,1GRP00950
     1)*AJ(3,2)-AJ(3,1)*AJ(2,2)*AJ(1,3)-AJ(3,2)*AJ(2,3)*AJ(1,1)-AJ(3,3)*GRP00960
     2AJ(2,1)*AJ(1,2)                                                   GRP00970
      IF(N.EQ.1) GO TO 130                                              GRP00980
      GO TO (70,80,90,70,80,90),NFACE                                   GRP00990
   70 CRP(1)=AJ(2,2)*AJ(3,3)-AJ(2,3)*AJ(3,2)                            GRP01000
      CRP(2)=AJ(2,3)*AJ(3,1)-AJ(2,1)*AJ(3,3)                            GRP01010
      CRP(3)=AJ(2,1)*AJ(3,2)-AJ(2,2)*AJ(3,1)                            GRP01020
      GO TO 100                                                         GRP01030
   80 CRP(1)=AJ(1,3)*AJ(3,2)-AJ(1,2)*AJ(3,3)                            GRP01040
      CRP(2)=AJ(1,1)*AJ(3,3)-AJ(1,3)*AJ(3,1)                            GRP01050
      CRP(3)=AJ(1,2)*AJ(3,1)-AJ(1,1)*AJ(3,2)                            GRP01060
      GO TO 100                                                         GRP01070
   90 CRP(1)=AJ(1,2)*AJ(2,3)-AJ(1,3)*AJ(2,2)                            GRP01080
      CRP(2)=AJ(1,3)*AJ(2,1)-AJ(1,1)*AJ(2,3)                            GRP01090
      CRP(3)=AJ(1,1)*AJ(2,2)-AJ(1,2)*AJ(2,1)                            GRP01100
  100 CONTINUE                                                          GRP01110
      SIGN=+1.0D0                                                       GRP01120
      IF(NFACE.GT.3) SIGN=-1.0D0                                        GRP01130
      DO 120 I=1,12                                                     GRP01140
      I1=NC(NFACE,I)                                                    GRP01150
      W3(I)=W2(I1)                                                      GRP01160
      DO 120 J=1,3                                                      GRP01170
  120 AK2(I,J)=CRP(J)*W3(I)*SIGN                                        GRP01180
      GO TO 150                                                         GRP01190
  130 CONTINUE                                                          GRP01200
      DO 140 L=1,NPEL                                                   GRP01210
  140 AK(L)=DTJ*W2(L)                                                   GRP01220
  150 CONTINUE                                                          GRP01230
      RETURN                                                            GRP01240
      END                                                               GRP01250
```

```
      SUBROUTINE GRID(NX,NY,NZ,NUMNP,NPL)                                MAP00010
      IMPLICIT REAL*8(A-H,O-W)                                           MAP00020
      COMMON /B2/ COORD(1050,3),COREL(32,3)                              MAP00030
      DIMENSION X(100),Y(100)                                            MAP00040
      REAL*8 LABEL/8H        /                                           MAP00050
      REAL*8 ITITLE(12)                                                  MAP00060
 1000 FORMAT(6A8)                                                        MAP00070
      IWIDE=9                                                            MAP00080
      IHIGH=15                                                           MAP00090
      IXUP=15                                                            MAP00100
      NY1=NY+1                                                           MAP00110
      NX1=NX+1                                                           MAP00120
      NZ1=NZ+1                                                           MAP00130
      NY2=3*NY+1                                                         MAP00140
      NX2=3*NX+1                                                         MAP00150
      NY3=5*NY+3                                                         MAP00160
      VXMAX=-1.0D+10                                                     MAP00170
      VYMAX=-1.0D+10                                                     MAP00180
      VXMIN= 1.0D+10                                                     MAP00190
      VYMIN= 1.0D+10                                                     MAP00200
      DO 100 I=1,NUMNP                                                   MAP00210
      VXMAX=DMAX1(VXMAX,COORD(I,1))                                      MAP00220
      VYMAX=DMAX1(VYMAX,COORD(I,2))                                      MAP00230
      VXMIN=DMIN1(VXMIN,COOR4(I 1))                                      MAP00240
  100 VYMIN=DMIN1(VYMIN,COORD(I,2))                                      MAP00250
      DO 200 I=1,NUMNP                                                   MAP00260
      COORD(I,1)=COORD(I,1)-VXMIN                                        MAP00270
  200 COORD(I,2)=COORD(I,2)-VYMIN                                        MAP00280
      XSCALE=((VXMAX-VXMIN)/9.0)*1.0E+10                                 MAP00290
      YSCALE=((VYMAX-VYMIN)/15.0)*1.0E+10                                MAP00300
      IF(XSCALE.LT.YSCALE) GO TO 300                                     MAP00310
      IHIGH=9                                                            MAP00320
      IXUP=9                                                             MAP00330
  300 IF(XSCALE.LE.YSCALE) XSCALE=YSCALE                                 MAP00340
      IF(XSCALE.GT.YSCALE) YSCALE=XSCALE                                 MAP00350
      IEXP=ALOG10(XSCALE)                                                MAP00360
      XSCALE=(XSCALE+(10**IEXP)*.5)*1.0E-10                              MAP00370
      DO 700 L1=1,NZ1                                                    MAP00380
      READ(5,1000) (ITITLE(I),I=1,6)                                     MAP00390
      READ(5,1000) (ITITLE(I),I=7,12)                                    MAP00400
      DO 500 I=1,NY1                                                     MAP00410
      I1=3*I-2+(L1-1)*NPL                                                MAP00420
      DO 400 J=1,NX2,3                                                   MAP00430
      J1=I1+NY3*(J-1)/3                                                  MAP00440
      X(J)=COORD(J1,2)                                                   MAP00450
      Y(J)=-COORD(J1,1)                                                  MAP00460
```

```
      IF(J.EQ.NX2) GO TO 400                                            MAP00470
      J2=J1+NY2-2*(I-1)                                                 MAP00480
      J3=J2+NY1                                                         MAP00490
      J4=J+1                                                            MAP00500
      J5=J4+1                                                           MAP00510
      X(J4)=COORD(J2,2)                                                 MAP00520
      Y(J4)=-COORD(J2,1)                                                MAP00530
      X(J5)=COORD(J3,2)                                                 MAP00540
      Y(J5)=-COORD(J3,1)                                                MAP00550
  400 CONTINUE                                                          MAP00560
      MODCUR=2                                                          MAP00570
      IF(I.EQ.1) MODCUR=1                                               MAP00580
  500 CALL DRAW(NX2, X, Y,MODCUR,0,LABEL,ITITLE,XSCALE,XSCALE,IXUP,     MAP00590
     10,2,2,IWIDE,IHIGH,0,LAST)                                         MAP00600
      DO 700 I=1,NX1                                                    MAP00610
      I1=(I-1)*NY3+(L1-1)*NPL                                           MAP00620
      DO 600 J=1,NY2,3                                                  MAP00630
      J1=I1+J                                                           MAP00640
      X(J)=COORD(J1,2)                                                  MAP00650
      Y(J)=-COORD(J1,1)                                                 MAP00660
      IF(J.EQ.NY2) GO TO 600                                            MAP00670
      J2=J1+1                                                           MAP00680
      J3=J2+1                                                           MAP00690
      J4=J+1                                                            MAP00700
      J5=J4+1                                                           MAP00710
      X(J4)=COORD(J2,2)                                                 MAP00720
      Y(J4)=-COORD(J2,1)                                                MAP00730
      X(J5)=COORD(J3,2)                                                 MAP00740
      Y(J5) =-COORD(J3,1)                                               MAP00750
  600 CONTINUE                                                          MAP00760
      IF(I.EQ.NX1) MODCUR=3                                             MAP00770
  700 CALL DRAW(NY2, X, Y,MODCUR,0,LABEL,ITITLE,XSCALE,XSCALE,IXUP,     MAP00780
     10,2,2,IWIDE,IHIGH,0,LAST)                                         MAP00790
      RETURN                                                            MAP00800
      END                                                               MAP00810
```

REFERENCES

1. Akyuz, F.A., "Natural Coordinate Systems: An Automatic Input Data Generation Scheme for a Finite-Element Method," Nuclear Engineering and Design, Vol. 11, No. 2, pp 195 - 207, March 1970. '69

2. Becker, E.B. and Brisband, J.J., Rohm and Haas Company, Huntsville, Alabama, Technical Report S-76, Application of the Finite Element Method to Stress Analysis of Solid Propellent Rocket Grains, Vol. 1, November 18, 1965, Vol. 2, pt 1, January 21, 1966.

3. Cantin, G., "Rigid Body Motions in Curved Finite Elements," AIAA Journal, Vol. 8, No. 7, pp 1252 - 1255, July, 1970.

4. Cantin, G., "Three-Dimensional Finite Element Studies," (Naval Postgraduate School report in preparation).

5. Dailey, G., The Johns Hopkins University Technical Memorandum TG 1128, A Computer Program to Plot Contour Maps of Stresses in Two-Dimensional Structures Analyzed by the Method of Finite Elements, July, 1970.

6. Dailey, G., The Johns Hopkins University Technical Memorandum TG 1140, Computer Programs for Plotting Finite-Element Patterns of Two- and Three-Dimensional Structures, October, 1970.

7. Holand, I. and Bell, K., Finite Element Methods in Stress Analysis, Chapter 13 by Zienkiewicz, Irons, Ergatoudis, Ahmad, and Scott, pp 383 - 435, Tapir, Technical University of Norway, Trondheim, Norway, 1969.

8. Jones, R.L., National Aeronautics and Space Administration Technical Note D - 6022, A Generalized Digital Contouring Program, January, 1971.

9. Sprague, R. L., National Aeronautics and Space Administration Technical Memorandum X - 2204, GIG - A Fortran IV Geometry Input Generator Program for a Finite-Element Heat-Transfer Code, March, 1971.

10. Serpanos, J.E., Naval Weapons Center, China Lake, California Technical Note 4062 - 47, SAGPAC - Structural Analysis Graphics Package, A Fortran IV Computer Program, June, 1970.

11. Stroud, A.H. and Secrest, D., Gaussian Quadrature Formulas, Prentice-Hall Inc., 1966.

12. Users Manual, W.R. Church Computer Center, U.S. Naval Postgraduate School, Monterey, California, 1970.

13. Zienkiewicz, O.C. and Cheung, Y.K., The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill Publishing Company, Ltd., London, 1967.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor Gilles Cantin, Code 59 Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	5
4.	Professor Robert E. Newton, Code 59 Ne Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Assistant Professor David Salinas, Code 59 Zc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	LT Dale E. Hanson, USN U.S. Naval Ship Repair Facility Box 34 FPO San Francisco 96651	1
7.	LT Emmanuel Leonidas SMC 1335 Naval Postgraduate School Monterey, California 93940	1
8.	LT Charles G. Pfeifer, USN SMC 1346 U.S. Naval Postgraduate School Monterey, California 93940	1
9.	Mr. D. Talapatre Department of Mechanical Engineering University of British Columbia Vancouver 8, Canada	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Three-Dimensional Elasto-Static Problems Using Isoparametric Finite Elements

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; June 1971

5. AUTHOR(S) (First name, middle initial, last name)

Dale E. Hanson

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1971	93	13

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

The objective of the project described in this report was to develop a computer system which would generate the required input data for a structural analysis of three-dimensional elasto-static problems using isoparametric finite elements. Element connectivity, nodal point coordinates, consistent gravity loads, and consistent pressure loads are generated. A variety of algorithms are used in the system to reduce the amount of input data required. The computer system and a sample problem are discussed.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31406

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Finite Element Mesh Generator						

28 27 MAR 72 19537
23 JUN 72 20468
16 APR 73 20901
5 AUG 80 27087

Thesis 128066
H2043 Hanson
c.1 Three dimensional
elasto-static problems using isoparametric finite elements.